

Natarajan Meghanathan
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

6th International Conference on Computer Science, Engineering and Information
Technology (CSEIT-2019)
November 23 ~ 24, 2019, Zurich, Switzerland



AIRCC Publishing Corporation

Volume Editors

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

Dhinaharan Nagamalai,
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-09-1
DOI: 10.5121/csit.2019.91301- 10.5121/csit.2019.91330

This work is subject to copyright. All rights are reserved, whether whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the International Copyright Law and permission for use must always be obtained from Academy & Industry Research Collaboration Center. Violations are liable to prosecution under the International Copyright Law.

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 6th International Conference on Computer Science, Engineering and Information Technology (CSEIT-2019) November 23 ~ 24, 2019, Zurich, Switzerland, International Conference on Machine Learning & Applications (CMLA 2019), 11th International Conference on Networks & Communications (NeTCoM 2019), International Conference on Internet of Things (CIoT 2019), 6th International Conference on Signal, Image Processing and Multimedia (SPM 2019), 11th International Conference on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC - 2019), 11th International Conference on Wireless & Mobile Networks (WiMoNe-2019), International Conference on Network and Communications Security (NCS 2019) was collocated with 6th International Conference on Computer Science, Engineering and Information Technology (CSEIT-2019). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CSEIT 2019, CMLA 2019, NeTCOM 2019, CIoT 2019, SPM 2019, Graph-hoc 2019, WiMoNe 2019 and NCS 2019 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, CSEIT 2019, CMLA 2019, NeTCOM 2019, CIoT 2019, SPM 2019, Graph-hoc 2019, WiMoNe 2019 and NCS 2019 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the CSEIT 2019, CMLA 2019, NeTCOM 2019, CIoT 2019, SPM 2019, Graph-hoc 2019, WiMoNe 2019 and NCS 2019.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

Natarajan Meghanathan
Dhinaharan Nagamalai (Eds)

General Chair

Natarajan Meghanathan
Dhinaharan Nagamalai (Eds)

Organization

Jackson State University, USA
Wireilla Net Solutions, Australia

Program Committee Members

Abdellatif I. Moustafa	Umm AL-Qura University, Saudi Arabia
Abilash	D.E & F.O Engineering, India
Ahmad A. Saifan	Yarmouk university, Jordan
Ali Abdrhman Mohammed Ukasha	Sebha University, Libya
Amizah Malip	University of Malaya, Malaysia
Anand Nayyar	Duy Tan University, Vietnam
Ankur Singh Bist	KIET Ghaziabad, India
Atanu Nag	Modern Institute of Engineering & Technology, India
Bouchra Marzak	Hassan II University, Morocco
Chandrashekhara Bhat	Manipal Institute of Technology (MIT), India
Deepali Gupta	Maharishi Markandeshwar University, Ambala, India
Grigorios N. Beligiannis	University of Patras, Greece
Gurjot Singh Gaba	Lovely Professional University, India
Hala Abukhalaf	Palestine Polytechnic University, Palestine
Hamid Ali Abed AL-Asadi	Basra University, Iraq
Hashem H. M. Ramadan	Technology Integration Engineer, India
HlaingHtakeKhaungTin	University of Computer Studies, Myanmar
Ilham Huseyinov	Istanbul Aydin University, Turkey
Iyad Alazzam	Yarmouk University, Jordan
Jonah Lissner	technion - israel institute of technology, Israel
Klenilmar Dias	Instituto Federal do Amapa/Universidade Federal de Minas Gerais, Brazil
Kfir Bar	College of Management, Israel
Liana Stanescu	University of Craiova, Romania
Muneer Masadeh Bani Yassein	Jordan University of Science and Technology, Jordan
Nahlah Shatnawi	Yarmouk University, Jordan
Narendra V G	Manipal Institute of Technology, India
Naresh Kumar Reddy	ICFAI Tech, India
Natarajan Meghanathan	Jackson State University, USA
Nesreen Alsharman	Isra university, Jordan
Nesrine Hafiene	MARS Research Laboratory, Tunisia
Osama Rababah	The University of Jordan, Jordan
Ouassila Hioual	Abbes Laghrour University, Algeria

Prabhat Mahanti	University of New Brunswick, Canada
Medve Anna	University of Pannonia, Hungary
Mohamed Ismail Roushdy	Ain Shams University, Egypt
Muhammad Shahzad Aslam	Xiamen University, Malaysia
Mohammed Awad	Arab American University, Palestine
Neda Darvish	Islamic Azad University, Iran
Nidal M. Turab	Al-Isra University, Jordan
Payal Chaudhari ,	Gujarat Technological University, India
Peiman Mohammadi	Islamic Azad University, Iran
Prasad S.Halgaonkar	MIT College of Engineering, India
Rafa E.Al-Qutaish	ETS - University of Quebec, Canada
Rajarajan M	City University, United Kingdom
Reza Ebrahimi Atani	University of Guilan, Iran
Rachid LATIF	Ibn Zohr University, Morocco
RHATTOY	Moulay Ismail University, Morocco
Ramgopal Kashyap	Amity University Chhattisgarh, India
Sabina Rossi	Università Ca' Foscari Venezia, Italy
Satria Mandala	Maliki Islamic State University (UIN Malang), Indonesia
Selçuk HELHEL	Akdeniz University, Turkey
Solomiia Fedushko	Lviv Polytechnic National University, Ukraine
Somayeh Mohammadi	Islamic Azad University, Iran
Subrata Dutta	Jadavpur University, India
Surekha Kamath	MIT, India
Subba Reddy	Manipal University, India
Ugur Alper	Pamukkale University, Turkey
Xiao-Zhi Gao	University of Eastern Finland, Finland
Yenke Blaise Omer	University Institute of Technology, Cameroon

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)



Artificial Intelligence Community (AIC)



Soft Computing Community (SCC)



Digital Signal & Image Processing Community (DSIPC)



Organized By



Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

6th International Conference on Computer Science, Engineering and Information Technology (CSEIT-2019)

Semantic Document Classification Based on Strategies of Semantic Similarity Computation and Correlation Analysis	01 - 17
<i>Shuo Yang, Ran Wei, Hengliang Tan and Jiao Du</i>	
Secure and Privacy-Aware Data Collection Architecture Approach in Fog Node Based Distributed IoT Environment.....	19 - 32
<i>Moussa WITTI and Dimitri KONSTANTAS</i>	
Quality Model to the Adaptive Guidance.....	33 - 43
<i>Hamid Khemissa and Mourad Oussala</i>	
A Hybrid Model for Evacuation Simulation and Efficiency Optimization in Large Complex Buildings.....	45 - 56
<i>Hao Yuan, Guo Yu, Yifan Ma, Jieneng Chen and Xiongda Chen</i>	
Collaborative and Fast Decryption Using Fog Computing and a Hidden Access Policy.....	57 - 71
<i>Ahmed Saidi, Omar Nouali and Abdelouahab Amira</i>	
Upgrading Cloud Infrastructure – Challenges and Solutions.....	73 - 82
<i>Andrei Petrescu and Mihai Carabas</i>	
An Intelligent Mobile Application to Manage College Database and Recommendation using Data Mining.....	83 - 89
<i>Yixuan Qi, Qi Lu, Yu Su and Fangyan Zhang</i>	
Automated Generation of Computer Graded Unit Testing-Based Programming Assessments for Education	91 - 100
<i>Sébastien Combéfis and Guillaume de Moffarts</i>	
SFERAnet: Automatic Generation of Football Highlights	101 - 116
<i>Vincenzo Scotti, Licia Sbattella and Roberto Tedesco</i>	
Automation Regression Testing for Sas.Am Website.....	117 - 137
<i>Harutyun Berberyan and Shahid Ali</i>	

TRIT: A Robust Tracker Based on Triplet Network..... 139 - 152
Peng Zou and Yunfei Cai

**Customized Garment Fashion Recommendation System using Data Mining
Techniques.....** 373 - 385
Shukla Sharma, Ludovic Koehl, Pascal Bruniaux and Xianyi Zeng

International Conference on Machine Learning & Applications (CMLA 2019)

Prediction and Causality Analysis of Churn Using Deep Learning..... 153 - 165
Muzaffar Shah, Darshan Adiga, Shabir Bhat and Viveka Vyeth

**An Artificial Neural Network Approach for the Classification of Human Lower
Back Pain.....** 167 - 172
Shubham Sharma and Rene V.Mayorga

**Cognitive Cities an Architectural Framework for the Cities
of the Future** 173 - 182
*Cristiana Carvalho, Filipe Cabral Pinto, Isabel Borges, Gonçalo Machado and Ilídio
Oliveira*

Automated Music Making with Recurrent Neural Network..... 183 - 188
You Peng, Ariel Jiang and Qi Lu

**Prediction of Workpiece Quality: an Application of Machine Learning in
Manufacturing Industry** 189 - 202
*Günther Schuh, Paul Scholz, Sebastian Schorr, Durmus Harman, Matthias Möller, Jörg
Heib and Dirk Bähre*

**A Facial Recognition-Based Video Encryption Approach to Prevent
Fakedeep Videos.....** 203 - 208
Alex Liang, Yu Su and Fangyan Zhang

**A Novel Machine Learning System for Sentiment Analysis
and Extraction** 387 - 393
Osama Mohammad Rababah and Nour Alokaily

11th International Conference on Networks & Communications (NeTCoM 2019)

Token Bucket-based Throughput Constraining in Cross-layer Schedulers.. 209 - 219
Jeremy Van den Eynde and Chris Blondia

Measurement and Characterization of the Stationary Noise in Narrowband Power Line Communication..... 221 - 232
Raja Alaya and Rabah Attia

International Conference on Internet of Things (CIoT 2019)

Educational Approach to the Internet of Things (IoT) Concepts and Applications 233 - 247
Rajeev Kanth, Tuomas Korpi, Arto Toppinen, Kimmo Myllymäki, Jatin Chaudhary and Jukka Heikkonen

Security Framework for IoT Devices Against Cyber-Attacks..... 249 - 266
Aliya Tabassum and Wadha Lebda

Hybrid Application Layer Protocol Design for IoT Environments..... 267 - 286
Erdal ÖZDOĞAN and O.Ayhan ERDEM

6th International Conference on Signal, Image Processing and Multimedia (SPM 2019)

Segmentation of Single and Overlapping Leaves by Extracting Appropriate Contours 287 - 300
Rafflesia Khan and Rameswar Debnath

Split Multi-Stage Vector Quantization based Steganography for Secure Wideband Speech Coder 301 - 312
Merouane BOUZID and Bakkar LASKAR

Lane Detection for Prototype Autonomous Vehicle 313 - 320
Sertap Kamçı, Dogukan Aksu and Muhammed Ali Aydin

**11th International Conference on Applications of Graph Theory in
Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC - 2019)**

**Finding Maximal Localizable Region in Wireless Sensor Networks by Merging
Rigid Clusters** 321 - 330
Saroja Kanchi

**11th International Conference on Wireless & Mobile
Networks (WiMoNe-2019)**

**Methodology to Evaluate WSN Simulators: Focusing on Energy Consumption
Awareness** 331 - 351
*Michel Bakni, Luis Manuel Moreno Chacon, Yudith Cardinale, Guillaume Terrasson and
Octavian Curea*

**11th International Conference on Network and Communications
Security (NCS 2019)**

**Public-Key based Authentication Architecture for IoT
Devices Using PUF** 353 - 371
Haji Akhundov, Erik van der Sluis, Said Hamdioui and Mottaqiallah Taouil

SEMANTIC DOCUMENT CLASSIFICATION BASED ON STRATEGIES OF SEMANTIC SIMILARITY COMPUTATION AND CORRELATION ANALYSIS

Shuo Yang^{1*}, Ran Wei², Hengliang Tan¹, and Jiao Du¹

¹School of Computer Science and Cyber Engineering, Guangzhou University,
Guangzhou, China

²Department of Computer Science, University of California, Irvine, California,
USA

ABSTRACT

Document (text) classification is a common method in e-business, facilitating users in the tasks such as document collection, analysis, categorization and storage. Semantic analysis can help to improve the performance of document classification. Though having been considered when designing previous methods for automatic document classification, more focus should be given to semantics with the increase number of content-rich electronic documents, forum posts or blogs online, which can reduce human workload by a great margin. This paper proposes a novel semantic document classification approach aiming to resolve two types of semantic problems: (1) polysemy problem, by using a novel semantic similarity computing strategy (SSC) and (2) synonym problem, by proposing a novel strong correlation analysis method (SCM). Experiments show that our strategies can help to improve the performance of the baseline methods.

KEYWORDS

semantic document classification, semantic similarity, semantic embedding, correlation analysis, machine learning

1. INTRODUCTION

Automatic document classification has many applications in numerous electronic business (e-business) scenarios [2]. For example, a medium-sized company may receive quite a few emails daily without accurate and concrete information such as recipient's name or department, which have to be read by an assigned agent so that the destinations can be determined. Thus, it is no doubt that an automatic document classification system can reduce human workload to a great extent.

More generally, given the rapid growth of web digital documents, it is often beyond one's ability to categorize information by reading thoroughly the pool of documents. Accurate and automatic text classification techniques are hence needed, which can classify the incoming text documents

into different categories such as news, emails, contracts, reports, etc. Users can hence estimate the content and determine the priorities of each document, maintaining more organized working schedule and creating more business value [28].

A quantitative definition of text classification was proposed by Aggarwal and Zhai [1]: given a set of text documents $D = \{x_1, x_2, \dots, x_N\}$, each document x_i has to be assigned with a set of different selected indices $\{1, 2, \dots, k\}$ that represents k different labels of text categories from an overall index list.

A typical method of automatic text classification is that given a training set of documents with known category labels and word dependency information, calculation on each member of the test document set has to end up with a list of possibilities on each label assigned to it. Certainly, the label with the highest likelihood corresponds to the predicted category that a test document belongs to. Classical machine learning (ML) algorithms such as Bayesian classifier, decision Tree, K-nearest neighbour, support vector machine and neural network were often applied in text classification [16]. In recent years deep learning algorithms are also introduced in these tasks. One of the representative trials was the application of convolutional neural network (CNN), a powerful network in computer vision [17]. Recurrent neural network, which has memory function that can capture sequence-formed information, was later introduced and became popular to handle classification problems [36].

However, most baselines mentioned above seldom view the classification problem from the perspective of semantic analysis. For example, the traditional Bayesian-based text classification method constructs a classification model based on the frequencies of some feature words in corpus. Unfortunately, this method does not take into account polysemous words (a word which holds different meanings depending on the context) and synonymous words (different words which hold a similar meaning) for semantic analysis during the classification procedure. For example, the Chinese word “Xiaomi” can mean either an agricultural product or a high-tech company; therefore, when classifying documents based on the traditional Bayesian method, documents including “Xiaomi” may be classified as “agriculture” or “technology”. Similar problems also exist in the classification of English documents. For example, English documents containing the word “program” may not only represent computer code programs and be classified as “computer”, but also represent a scheduled radio or television show and be classified as “entertainment”. Similarly, English articles containing the word “center” can either represent a geometric center and be classified as “mathematics”, or an important place of economy and culture and be classified as “geography”.

On the other hand, synonymous words can also cause mis-classification of documents. For example, the word “people” is synonymous with “mass” and “mob”. But they may occur in articles about different topics (e.g., architecture, culture and history). Therefore, choosing these words as features of the classification model may cause classification errors. These situations also exist in document classification tasks of word-embedding-based deep learning methods. For example, during feature extraction procedure the word dependence is calculated based on network training upon a particular corpus; in other words, the result is based on the statistical analysis on the posterior probability of a word following another one. However, a single embedding cannot represent multiple meanings, while similar embeddings may refer to different topic types.

The above issues can be summarized as two research problems:

(1) Problem of polysemy: some words have multiple meanings, which may lead to mis-classification of documents;

(2) Problem of synonym: different words with similar meanings are often used in different scenarios, but when they appear in an article at the same time, it may lead to mis-classification of documents;

Khan et al. [16] suggested that semantic analysis could help enhance the performance of classification. Previous work has made significant progress on this task. Fang, Guo, Wang and Yang (2007), and Khan, Baharudin, Lee and Khan (2010) claimed that semantic analysis can be generally implemented by the introduction of ontology that represents terms and concepts in a domain-wised manner [8,16]. However, ontologies are particularly pre-defined domain-constraint expert knowledge base. They are not good at eliminating ambiguity across different fields (domains) or different natural languages, which may lead to polysemy and synonymy issues [29], finally resulting in uncertainty of document classification [9]. Liu, Scheuermann, Li and Zhu (2007) proposed a text classification method based on WordNet for word sense disambiguation (WSD) [20]. Some other approaches use supervised (Jin, Zhang, Chen, Xia (2016) [12]) or unsupervised method or the joint method of them (Wawer and Mykowiecka (2017) [32]) for word disambiguation. However, few methods consider document misclassification caused by both the ambiguity of polysemy and multi-scene characteristics of synonym at the same time. In recent years, some approaches use name entities for text classification. For example, Stefan, Miroslav, Ivan, Marko and Aleksandar (2017) proposed a method based on name entity network linking. However, the author showed that their experiment results did not show any significant improvement when using named entities, and in some cases even worse performance [28]. Türker, Koutraki, Zhang and Sack (2018) proposed an approach based on a name entity dictionary (i.e., Anchor-Text Dictionary). However, if the words of the text do not exist in the dictionary, the classification results may be biased [30].

HIT IR-Lab Tongyici Cilin (Extended) proved that extending word meaning effectively or replacing keywords with synonyms can significantly improve the performance of information retrieval, text classification and automatic question answering system [13]. Motivated by this linguistic evidence, in this research, we propose two strategies to improve the performance of semantic document categorization of baselines. The *first* strategy aims to solve polysemy problem by using a novel semantic similarity computing method (SSC) so that the most context-fitting meaning of a word can be determined by referring to the meaning of similar sentences in a common dictionary. In this paper, *CoDic* [10,34] and *Hownet* [7] are used as common dictionaries for meaning determination and term expansion. With the help of *CoDic* and *Hownet*, words with ambiguity will be removed from the feature list, enabling more distinctive features to be selected. The *second* strategy aims to solve the synonym problem by adopting a strong correlation analysis method (SCM), where synonyms unrelated to the classification task are deleted. Otherwise, select the specific meaning of one word in the synonym group from the common dictionary and replace the other synonyms in the same group.

2. RELATED WORK

Automated document classification, also called categorization of document, has a history that can date back to the beginning of the 1960s. The incredible increase in online documents in the last decades intensified and renewed the interests in automated document classification and data mining. In the beginning, document classification focused on heuristic methods, that is, solving the task by applying a group of rules based on expert knowledge. However, this method was proved to be inefficient, so in recent years more focuses are turned to automatic learning and clustering approaches. These approaches can be divided into three categories based the characteristics of their learning phases:

(1) *Supervised document classification*: this method guidelines the whole learning process of classifier model by providing complete training dataset that contains document content and category labels at the same time. The process of supervision is like that of students doing exercises which have correct answers for them to refer to.

(2) *Semi-supervised document classification*: a mixture method between supervised and unsupervised document classification. Parts of documents have category labels while the others do not.

(3) *Unsupervised document classification*: this method is executed without priori knowledge of the document categories. The process of unsupervised learning is like that of students doing final examination which they do not have standard answers for reference.

However, no matter what kinds of learning methods, many of them require to firstly convert unstructured text to digital numbers in the data pre-processing stage. The most traditional (and intuitional) algorithm is one-hot representation, which uses N-dimension binary vector to represent vocabulary with each dimension stands for one word [16]. However, this strategy easily incurs curse of dimensionality for representation of long texts. This is because a big vocabulary generates high-dimension, but extremely sparse vectors for long documents. Therefore, dimensionality reduction operation which removes redundant and irrelevant features is needed [4]. This demand is satisfied by the methodology called feature extraction/selection. The goal of feature extraction is the division of a sentence into meaningful clusters and meanwhile removing insignificant components as much as possible. Typical tasks at the pre-processing stage include tokenization, filtering, lemmatization and stemming [31]. After that, feature selection aims to select useful features of a word for further analysis. Compared with one-hot representation that generates high-dimensional, sparse vectors, an improved solution called TF-IDF produces more refined results. In this frequency-based algorithm, the importance of a word is represented by the product of term frequency (how frequent the word shows up in a document) and inverse document frequency (log-inverse of the frequency that documents containing such word in the overall document base) [21,31]. These two algorithms, however, clearly suffer from limitations brought by neglecting the grammar and word relations in documents. More recently, distributed representation that illustrates dependencies between words are more widely used, as it reflects the relationships of words in one document [23]. Currently, the most widely used strategy to learn the vectorized words is to maximize the corpus likelihood (prediction-based), with the word2vec toolbox being one of the most popular tools. Implementation of this algorithm is dependent on the training of representation neural network with words in the form of binary vectors generated by

one-hot representation. The weights of the network keep being updated until convergence, which generates a vector that lists the possibility of each word could follow the input word in a document [16,23].

3. SEMANTIC DOCUMENT CLASSIFICATION

This section proposes two novel strategies to resolve the research problems mentioned above.

3.1. Strategy to Resolve Polysemy Problem: SSC

The first strategy aims to solve polysemy problem by using a novel semantic similarity computing method. The most context-fitting meaning of a word can be determined by referring to the semantics of related sentences in a common dictionary (e.g., CoDic for English and HowNet for Chinese).

In this strategy, we implement the semantic similarity computing method (*SSC*) for the similarity between two sentences. The *SSC* splits a text document into sentences. For each word (w) in a sentence (s), all of its concepts from the dictionary are extracted based on its Part-of-speech (*PoS*) tag in the sentence. Then, semantically compare each concept of w with s and return the concept with the maximum similarity score. Words that are not determinative of their exact meanings will be removed from the list of features, and hereby more distinctive terms are more likely to be selected as features. The pseudocode of the *SSC* algorithm is shown as Table 1.

The workflow of the *SSC* is quite simple. From Table 1, it is clear that the first step is to segment each sentence into words (*word_tokenize*) and tokenize each word (*pos_tag*) with its part of speech. Then, we get the synonym set (*synset*) for each tagged word in the sentence according to their PoS (*tagged_to_synset*). After that, we filter out the null component in each synset. Next, for each synset in the first sentence (*sent1*), we compute the similarity score of the most similar word (*compute_similarity*) in the second sentence (*sent2*). The aim of our function *compute_similarity* is to measure the similarity between two synsets. If two words are similar, their synsets should also be similar. This is because if two words are very similar, then their correlations with the same some other words will be very close. On the other hand, if the correlation between two words and the same some other words is close, then the two words are similar to each other [26].

Table 1. Semantic similarity computing (SSC)

Algorithm: semantic similarity computing (SSC)
Input: target sentence (ts); a set of test sentences (ss)
Output: the most similar sentence (s in ss) to ts with its maximum similar score (max)

```
def sentence_similarity(sentence1, sentence2)
  #Tokenize & pos tag
  sentence1 = pos_tag(word_tokenize(sentence1))
  sentence2 = pos_tag(word_tokenize(sentence2))
  # Get the synsets for the tagged words
  synsets1 = [tagged_to_synset(*tagged word) for tagged_word in sentence1]
  synsets2 = [tagged_to_synset(*tagged word) for tagged_word in sentence2]
  # Filter out the Null values
  synsets1 = [synset1 for synset1 in synsets1 if synset1]
```

```

synsets2 = [synset2 for synset2 in synsets2 if synset2]
score, count = 0.0, 0
# For each word in the first sentence
for synset1 in synsets1
# Get the similarity score of the most similar word in the second sentence
    best_score = max([synset1.compute_similarity(synset2) for synset2 in synsets2] )
# Check that whether the similarity could have been computed if best score is not None
    score += best_score
    count += 1
# Average the values
    score /= count
    return score # end of sentence similarity function
# __main__
    max = 0.0
    most_similar_sentence = None
    for s in ss
        value1 = sentence_similarity(s, ts)
        value2 = sentence_similarity(ts, s)
        avg_similarity = (value1 + value2) / 2
        if avg_similarity > maximum:
            most_similar_sentence = s
            max = avg_similarity
    print ("The most similar sentence is {}, with score {}".format(most_similar_sentence, max))

```

In the function of *compute_similarity*, when calculating the similarity of any two words in two synsets, we applied the mean value of multiple methods (if applicable): Path Similarity (PS) [3], Leacock-Chodorow (LCH) [18], Wu-Palmer (WUP) [33] and Lin [19]. This is because when using thesaurus (dictionary) alone to calculate the similarity, if the word is not in the dictionary, the similarity cannot be calculated.

PS computes the shortest number of edges from one word to another, assuming that a hierarchical structure exists (like WordNet that is essentially a graph) [22]. In general, two word that have a longer path distance are less similar than those with a very short path distance. If there is no path between two words, PS will return a Null value. This is another reason why we use different similarity measures.

$$\text{sim}_{\text{path}}(c_1, c_2) = \text{pathLen}(c_1, c_2) \quad (1)$$

where c_1, c_2 are two words, and $\text{pathLen}(c_1, c_2)$ is the shortest number of edges between those two words in a given thesaurus.

LCH is almost the same as PS, except it uses the negative logarithm of the result of the length of path.

$$\text{sim}_{\text{path}}(c_1, c_2) = -\log(\text{pathLen}(c_1, c_2)) \quad (2)$$

Based on LCH, WUP metric expands it by weighting the edges according to the distance in the hierarchy. Unlike the above methods, Lin metric considers similarity as both the information

content shared between two words, and the difference. It calculates the probability of the lowest common word between two words c_1 and c_2 , which is the lowest-leveled node in the hierarchy that is the parent of both c_1 and c_2 based on the corpora used [22].

After computing the similarity score of all synsets of sent1 with that of sent2, an average similarity value between them can be returned. By using this method, we can acquire the similarity values between all test sentences (ss) and the target sentence (ts). In the end, the test sentence with the maximum similarity value can be chosen as the most semantically similar sentence.

3.2. Strategy to Resolve Synonym Problem: SCM

There may be many synonyms in a large text, but not all of them are suitable as text features. As is known to all, selecting effective text features can reduce the dimension of feature space, enhance the generalization ability of the model and reduce overfitting, so as to improve the effect and efficiency of classification and clustering [5]. Therefore, effective feature selection is particularly important. In this section, we can turn the synonym problem into a sub-problem: how to determine the degree of the relevance between a feature and the classification task and then remove the feature words in the synonym group that are weakly relevant to or irrelevant to the classification task.

In this paper, a novel correlation analysis algorithm, named SCM, is proposed to obtain effective feature sets. The idea of the SCM contains two important considerations:

The feature words with strong category discrimination ability are extracted by using the category discrimination method (CDM), and then the correlation between other feature words and categories is measured by the feature correlation analysis (FCA). That is, the selected feature is guaranteed to be the most relevant to the category first, and then the degree of correlation between other features and selected features is calculated.

If a feature has a strong correlation with the selected feature, the SCM will not include it into the feature candidate set even if the feature has a strong correlation with the category. Because compared with existing feature candidate set, the new undetermined features cannot provide additional category-related information.

This paper adopts TF-IDF (Term Frequency-Inverse Document Frequency) [14,27] as the implementation of CDM. By applying TF-IDF to the synonym group in undetermined features, we can get a feature candidate set composed of a number of features with strong category discrimination ability. The TF-IDF method is a frequency-based algorithm. In TF-IDF, the importance of a word is represented by the product of the word frequency (i.e., the frequency with which the word appears in the document) and the inverse document frequency (i.e., dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient). The formulas of TFIDF are as follows.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (3)$$

$$\text{idf}_i = \lg \frac{|D|}{|\{D_j: t_i \in d_j\}| + 1} \quad (4)$$

$$\text{tf} - \text{idf}_{i,j} = \text{tf}_{i,j} * \text{idf}_i \quad (5)$$

where (3) refers to the importance of a term t_i in a particular document d_j . The molecule $n_{i,j}$ is the number of occurrences of t_i in d_j , and the denominator is the sum of the number of occurrences of all words in d_j . Formula (4) is a measurement of the general importance of a word in all documents. Its molecule represents the total number of documents in the corpus. The denominator represents the number of documents containing the word t_i . Formula (5) is the product of “term (word) frequency (TF)” and “inverse document frequency (IDF)”. The more important a word is to a certain category of texts, the higher its tf-idf value will be, and vice versa. Therefore, TF-IDF tends to filter out common words and retain important words to certain category of texts.

The SCM proceeds to calculate how strongly all features (in each synonym group) are related to category (C) in the feature candidate set. The formulas are as follows,

$$H(x) = \sum_{i=0}^n (p_i * \lg \frac{1}{p_i}) \quad (6)$$

$$H(X|Y) = \sum_j p(Y_j) \sum_i p(X_i|Y_j) \lg \frac{1}{P(X_i|Y_j)} \quad (7)$$

$$I(X|Y) = H(X) - H(X|Y) \quad (8)$$

$$\text{Corr}(X, Y) = \frac{I(X|Y) + I(Y|X)}{H(X) + H(Y)} \quad (9)$$

where X is an n-dimensional random variable and Y is a certain of class (or category). Formula (6) represents the entropy of X, that is the uncertainty of X. Formula (7) means the uncertainty of X given the occurrence of Y. Formula (8) represents information gain between $H(X)$ and $H(X|Y)$. Formula (9) is used to measure the degree of correlation between a feature (X) and a category (Y).

According to the degree of correlation, the features in each synonym group are arranged in a descending order respectively, and then the ordered feature sequences are put back into the feature candidate set. Select the first feature in the sequence, that is, the feature with the strongest correlation with category (C), and remove it from the feature candidate set and put it into the feature result set.

In order to eliminate redundant features, it is necessary to calculate the degree of mutual independence between any two features (within a synonym group). Thus, this section proposes a novel feature correlation analysis method, called FCA, to exclude unnecessary features in synonym groups of the feature candidate set. The idea of the FCA is simple: if a remaining feature in the candidate set is a strong category-correlated feature, and its mutual independence with the selected feature is greater than or equal to a threshold α , it indicates that the candidate feature is independent of the selected feature, and it needs to be included in the feature

result set. Otherwise, the feature is considered as redundant and should be deleted. Repeat this process until the feature candidate set is empty. The formulas are as follows:

$$\text{IDP}(X_i, Y|X_j) = \frac{I(X_i, Y|X_j) + I(X_j, Y|X_i)}{2H(Y)} \quad (10)$$

$$I(X; Y|Z) = \lg \frac{p(X|YZ)}{p(X|Z)} \quad (11)$$

where (10) is used to measure the degree of mutual independence between feature X_i and feature X_j when the category (Y) is known. Formula (11) describe the mutual information between feature X and feature Y in the case of given condition Z.

4. EXPERIMENTS

This section first introduces the datasets and evaluation metrics. Then, we experiment our strategies based on several baselines with detailed experimental procedure. After that, classification assessment is given based on the performance.

4.1. Dataset and Evaluation Metrics

To test the reliability and robustness of our strategy, we use:

Dataset 1: a movie review dataset from Rotten Tomatoes [24, 37]. This dataset contains 10662 samples of review sentences, with 50% positive comments and the remaining negative ones. The size of the vocabulary of the dataset is 18758. Since the dataset does not come with an official train/test split, we simply extract 10% of shuffled data as evaluation (dev) set to control the complexity of model. In the next research stage, we will use 10-fold cross-validation on the dataset.

Dataset 2: 56821 Chinese news dataset, which is available in PaddlePaddle¹ that is an open source platform launched by Baidu for deep learning applications. It contains 10 categories: international (4354), culture (5110), entertainment (6043), sports (4818), finance (7432), automobile (7469), education (8066), technology (6017), stock (3654) and real estate (3858). We assess the classification quality automatically with macro-average on accuracy and loss.

4.2. Experiment on neural network (NN)

In this experiment, the baseline CNN is taken as an example to compare the performance of classical NN and the improved one with our proposed strategy in document classification. The detail of model parameters is listed in Table 2. Both of the two trained models are evaluated on the *dev* dataset every 100 global steps and then they are stored in checkpoints before the training process starting again. After multiple training epochs, the models stored in checkpoint can be recovered and used for testing on a new dataset. Partial code for this work is available on github². The experimental procedure is described as follows.

(1) Each document in the corpus will be firstly transformed into our semantic document (i.e., documents with semantics embedding) [35] by extending each polysemous word and category-correlated synonymous word with its context-fitting concepts from the common dictionary (i.e., CoDic for English and Hownet for Chinese) with the help of the SSC and the SCM strategies, which aims for accurate semantic interpretation and term expansion.

CoDic is a semantic collaboration dictionary constructed under our CONEX project [10,34,35]. In CoDic, each concept is identified by a unique internal identifier (iid). The reason of this design is to guarantee semantic consistency and interoperability of documents while transferring across heterogeneous contexts. For example, from Figs. 1 and 2, it is clear that in CoDic, the word “program” with the meaning of “a scheduled radio or television show” is uniquely labelled by an iid “0x5107df021015”, while its another meaning “a set of coded instructions for insertion into a machine...” has another unique iid “0x5107df02101c”. Currently, CoDic is implemented in XML, where each concept is represented as an entry with a unique *iid* (see Fig. 3). It is convenient to extract all different meanings of any given word for later semantic analysis by using existed packages (e.g., *xml.etree.cElementTree* for Python and *javax.xml.parsers* for Java). Hownet as a common dictionary to handle Chinese documents is used similarly.

Table 2. Parameter settings of our experiments

Parameters	values
Percentage of splitting a dataset for training, testing and validating, respectively	0.8/ 0.1/ 0.1
Dimensionality of character embedding	128
Filter sizes	3,4,5
Number of filters per filter size	128
Dropout keep probability	0.5
L2 regularization lambda	0.01
Batch Size	64
Number of training epochs	1/ 5/ 10/ 50/ 100
Evaluate model on evaluation (dev) dataset after these steps	100

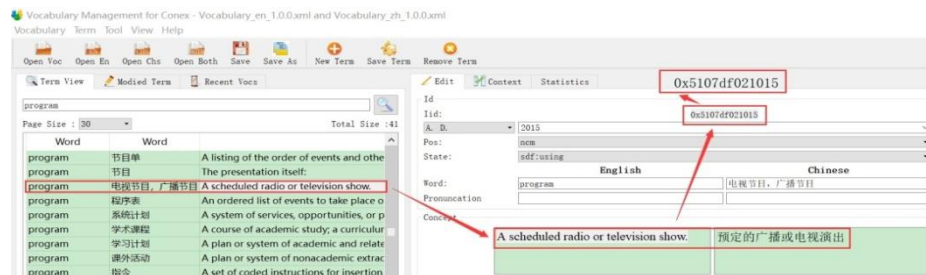


Figure 1. Word “program” with the meaning “a scheduled radio or television show” in CoDic

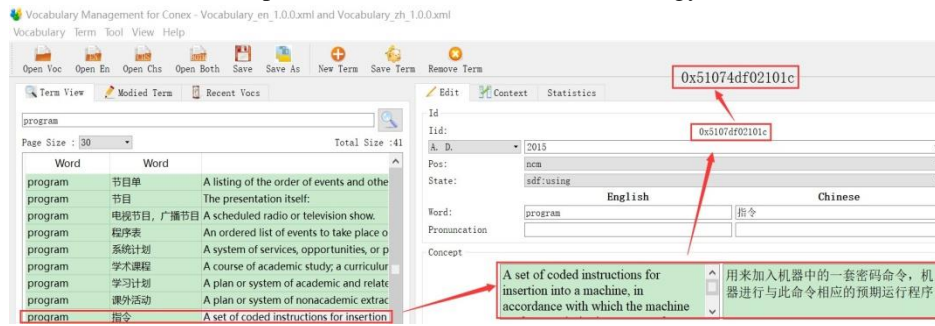


Figure 2. Word “program” with the meaning “a set of coded instructions for insertion into a machine” in CoDic

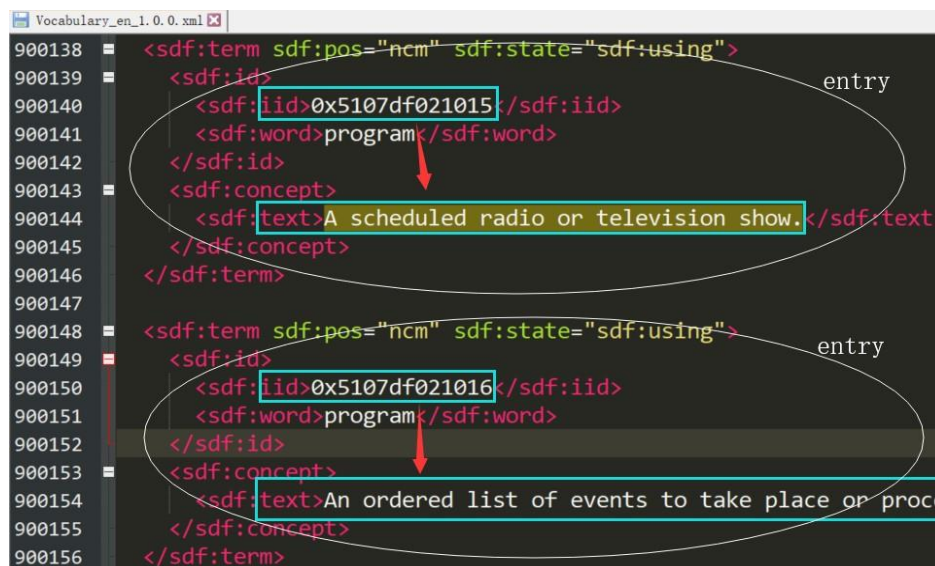


Figure3. CoDic in XML

(2) Build a Sem_{CNN} (CNN+SSC/SCM) network. The first layer embeds words and their extracted accurate concepts into low-dimensional vectors. The second layer performs convolutions over the semantic-embedded document tensors using different sized filters (e.g., $filter\ size = [3, 4, 5]$). Different sized filters will create different shaped feature maps (i.e., tensors). Third, max-pooling is used to merge the results of the convolution layer into a long feature vector. Next, dropout regularization is added in the result of max-pooling to trade-off between the complexity of the model being trained and the generalization of testing on evaluation dataset. The last layer is to classify the result using a Softmax strategy.

(3) Calculate loss and accuracy. The general loss function for classification problems is the cross-entropy loss which takes the prediction and the real value as input. Accuracy is another useful metric being tracked during training and testing processes. It can be used to prevent model overfitting during model training. At the beginning of the training, the training error on training dataset and the verification error on the evaluation dataset will decrease continuously. However, when the training process reaches a certain critical point, the accuracy of classification on the evaluation dataset will decline while the accuracy of training will continue to increase. At this

time, in order to avoid overfitting of the model, the training process should be interrupted and the parameters at the critical point should be used as the training results of the model.

(4) Record the summaries/checkpoints during training and evaluation. After an object declaration of *CNN/Sem_{CNN}* class, batches of data are generated and fed into it to train a reliable classification model. While the loss and accuracy are recorded to keep track of their evolvment over iterations, some important parameters (e.g., the embedding for each word, the weights in the convolution layers) are also needed to be saved for later usage (e.g., testing on new datasets).

(5) Test the classification model. Data for testing are loaded and their true labels are extracted for computing the performance of prediction. Then, the classification model is restored from the checkpoints, executing on the test dataset and producing a prediction for each semantic document. After that, the prediction results are compared with the true labels to obtain the testing accuracy of the classification model.

4.3. Experiment on ML approaches

The procedures of training classification models using classical machine learning algorithms with the proposed strategies are listed as follows, while the details can be also found in our open source code.

(1) Transform words into vectors based on inputted texts (Note: Chinese document needs to execute word segmentation beforehand.). Collect all words used in texts, perform a frequency distribution and then find out effective features suitable for document classification by using the proposed strategies (SSC and SCM). After that, each text will be converted to a long word vector, where True (or 1) means a word (or a feature) exists while False (or 0) means absent.

(2) Execute multiple classical machine learning approaches (e.g., Naïve Bayes, NB) based on the word vectors from Step (1). In this experiment, three variants of NB classifier are used. They are Original NB, multinomial NB and Bernoulli NB classifier. All of them take word features and corresponding category labels as input to train classification models. It is of note that sometimes the classifier should be modified based on realistic cases. For example, in order to avoid the probability being close to zero and underflow problem in NB, it is better to initialize the frequency of each word to one and take natural log of the product in the computation of posterior probability, respectively.

(3) Save the trained classifiers for later usage. This is because the training process might be time-consuming, which depends on numerous factors such as dataset size and the computation complexity during model training. Thus, it is impractical to train classification models each time while you need to use them.

(4) Boost multiple classifiers to create a voting system that is taken as a baseline for comparison. To do this, we build a typical classifier (i.e., *VoteClassifier*) with multiple basic classical ML classification algorithms (i.e., taking multiple basic classifier objects as input when initialized), each of which gets one vote. In *VoteClassifier*, the *classify* method is created by iterating through each basic ML classifier object to classify based on the same input features. This experiment chooses the most popular metrics (e.g., accuracy) among these classifiers. The classification can be regarded as a vote. After iterating all the classifier objects, it returns the most popular vote.

4.4. Experiment Result and Analysis

In the actual testing process, we need to maintain a common synonymous word dictionary and a common polysemous word dictionary. The reason we need to maintain these two dictionaries is that the computation workload to judge polysemy and synonyms in a long text are very heavy. For example, if there are n words in a text and each word has m different meanings, then the computational complexity of determining polysemous words is $O(n*m)$, and the computational complexity of determining synonyms is $O(n*(n-1))$, so that the total computational complexity is $O(n*(m+n-1)) > O(n^2)$. Therefore, maintaining these two dictionaries can reduce computational complexity and reduce the pre-processing time of text classification.

Table 3 shows the experimental comparison between classical machine learning algorithms and their improved counterparts on Dataset 1. In this experiment, classical machine learning algorithms include Original Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB), Logistic Regression (LR), support vector machine (SVM) with stochastic gradient descent (SGD), Linear SVC (SVC) and Nu-Support Vector Classification (NSVC).

From Table 3, it is clear that our improved algorithms have better performance than the classical ML algorithms in the accuracy of model prediction on the evaluation dataset. It is of note that three-variant NB algorithms and LR perform better than three-variant SVM algorithms, in both of the classical ones and improved ones. The VoteClassifier plays a role of baseline for the comparison between different algorithms. Table 4 and 5 show that Sem_{CNN} performs better than CNN in terms of accuracy and loss in different numbers of epochs. As the number of epoch increases, both of them increase in the accuracy of evaluation and decrease in the loss continuously (**before reaching overfitting**).

Table 3. Comparison of classical machine learning algorithms and our improved ones on Dataset 1

Accuracy (%)		Accuracy (%)	
NB	73.493	Improved NB	78.464
MNB	74.698	Improved MNB	79.518
BNB	74.096	Improved BNB	79.819
LR	73.494	Improved LR	76.506
SGD	69.879	Improved SGD	74.096
SVC	72.741	Improved SVC	73.946
NSVC	72.892	Improved NSVC	76.355
VoteClassifier	74.397	Improved VoteClassifier	74.398

Table 4. Comparison of SemCNN and traditional CNN on Dataset 1.

Number of epochs	Accuracy		Loss	
	<i>Sem_{CNN}</i>	CNN	<i>Sem_{CNN}</i>	CNN
Epoch = 1	0.586	0.568	0.818	0.876
Epoch = 5	0.713	0.676	0.567	0.59
Epoch = 10	0.744	0.722	0.519	0.62
Epoch = 50	0.841	0.724	0.621	0.742
Epoch = 100	0.902	0.739	0.961	0.999

Table 5. Comparison of SemCNN and traditional CNN on Dataset 2.

Number of epochs	Accuracy		Loss	
	<i>Sem_{CNN}</i>	CNN	<i>Sem_{CNN}</i>	CNN
Epoch = 1	0.861	0.828	0.473	0.560
Epoch = 5	0.956	0.923	0.211	0.295
Epoch = 10	0.990	0.953	0.095	0.212
Epoch = 20	0.990	0.966	0.0919	0.170

5. CONCLUSION

This paper introduces new strategies for semantic document classification. It mainly has two improvements: (1) solving polysemy problem by using a novel semantic similarity computing method (SSC). The SSC implements semantic analysis by executing semantic similarity computation and semantic embedding with the help of common dictionary. In this paper, we use CoDic for English texts and HowNet for Chinese texts. (2) solving synonym problem by proposing a novel strong correlation analysis method (SCM). The SCM consists of the CDM strategy for the selection of feature candidate set and the FCA strategy for the determination of the final feature set. Experiments show that our strategy can improve the performance of semantic document classification compared with that of traditional ones.

We will continue going deep in this research of semantic document classification. More multiple deep learning models (e.g., DualTextCNN, DualBiLSTM, DualBiLSTMCNN or BiLSTMAttention) will be tested for semantic document similarity on well-known document datasets with different natural languages. We would also try to compare our strategies with state-of-the-art embedding methods such as FastText [15], BERT [6] and ULMFit [11] and ELMo [25] and other classification methods such as the ones based on knowledge graph.

ACKNOWLEDGMENT

This research is supported by both National Natural Science Foundation of China (grant no.: 61802079 and 61802418) and Guangzhou University grant (no.: 2900603143). The authors would like to thank all the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

- [1] Aggarwal, C.C., Zhai, C.: Mining text data. Springer Science & Business Media (2012)
- [2] Altinel, B., Ganiz, M.C.: Semantic text classification: A survey of past and recent advances. *Information Processing & Management* 54(6), 1129–1153 (2018)
- [3] Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1), 13–47 (2006)
- [4] Cerda, P., Varoquaux, G., Kégl, B.: Similarity encoding for learning with dirty categorical variables. *Machine Learning* 107(8-10), 1477–1494 (2018)
- [5] Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Computers & Electrical Engineering* 40(1), 16–28 (2014)
- [6] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- [7] Dong, Z., Dong, Q., Hao, C.: HowNet and the computation of meaning (2006)
- [8] Fang, J., Guo, L., Wang, X., Yang, N.: Ontology-based automatic classification and ranking for web documents. In: *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*. vol. 3, pp. 627–631. IEEE (2007)
- [9] Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47(1), 1–66 (2017)
- [10] Guo, J., Da Xu, L., Xiao, G., Gong, Z.: Improving multilingual semantic interoperation in cross-organizational enterprise systems through concept disambiguation. *IEEE Transactions on Industrial Informatics* 8(3), 647–658 (2012)
- [11] Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
- [12] Jin, P., Zhang, Y., Chen, X., Xia, Y.: Bag-of-embeddings for text classification. In: *IJCAI*. vol. 16, pp. 2824–2830 (2016)
- [13] Jiu-le, T., Wei, Z.: Words similarity algorithm based on tongyici cilin in semantic web adaptive learning system [j]. *Journal of Jilin University (Information Science Edition)* 6(010) (2010)
- [14] Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* (2004)
- [15] Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
- [16] Khan, A., Baharudin, B., Lee, L.H., Khan, K.: A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology* 1(1), 4–20 (2010)
- [17] Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
- [18] Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49(2), 265–283 (1998)
- [19] Lin, D., et al.: An information-theoretic definition of similarity. In: *Icml*. vol. 98, pp. 296–304. Citeseer (1998)
- [20] Liu, Y., Scheuermann, P., Li, X., Zhu, X.: Using wordnet to disambiguate word senses for text classification. In: *international conference on computational science*. pp. 781–789. Springer (2007)
- [21] Manning, C.D., Raghavan, P., Schütze, H.: Scoring, term weighting and the vector space model. *Introduction to information retrieval* 100, 2–4 (2008)
- [22] Martin, J.H., Jurafsky, D.: *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River (2009)
- [23] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)
- [24] Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of the 43rd annual meeting on association for computational linguistics*. pp. 115–124. Association for Computational Linguistics (2005)

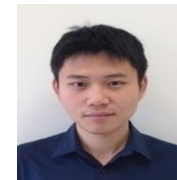
- [25] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
- [26] Qun, L., Sujian, L.: Semantic similarity calculation based on zhiwang. *International Journal of Computational Linguistics and Chinese Language Processing* 7(2), 59–76 (2002)
- [27] Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. Tech. rep., Cornell University (1982)
- [28] Stefan Aneli, Miroslav Kondi, I.P.M.J.A.K.: Text classification based on named entities. In: *ICIST*. pp. 23–28 (2017)
- [29] Thangaraj, M., Sivakami, M.: Text classification techniques: A literature review. *Interdisciplinary Journal of Information, Knowledge & Management* 13 (2018)
- [30] Tu`rker, R., Zhang, L., Koutraki, M., Sack, H.: Tecne: Knowledge based text classification using network embeddings. In: *EKAW (Posters & Demos)*. pp. 53–56 (2018)
- [31] Wang, Y., Wang, X.J.: A new approach to feature selection in text classification. In: *2005 International conference on machine learning and cybernetics*. vol. 6, pp. 3814–3819. IEEE (2005)
- [32] Wawer, A., Mykowiecka, A.: Supervised and unsupervised word sense disambiguation on word embedding vectors of unambiguous synonyms. In: *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*. pp. 120–125 (2017)
- [33] Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. pp. 133–138.
- [34] Association for Computational Linguistics (1994)
- [35] Xiao, G., Guo, J., Gong, Z., Li, R.: Semantic input method of chinese word senses for semantic document exchange in e-business. *Journal of Industrial Information Integration* 3, 31–36 (2016)
- [36] Yang, S., Wei, R., Shigarov, A.: Semantic interoperability for electronic business through a novel cross-context semantic document exchange approach. In: *Proceedings of the ACM Symposium on Document Engineering 2018*. p. 28. ACM (2018)
- [37] Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. *ieee Computational intelligence magazine* 13(3), 55–75 (2018)
- [38] Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)

AUTHORS

Shuo Yang received the Master's degree in software engineering from the Dalian Jiaotong University, China, in 2013. He was awarded a doctorate degree in software engineering, University of Macau, in 2017. Currently, he is a researcher in Guangzhou University. His research interests include semantic interoperability and semantic inference with AI technology, mainly applied to the fields of e-commerce, e-marketplace and clinical area.



Ran Wei received the Ph.D. degree in biomedical science from Rutgers University, USA, in 2018. He is currently a researcher in the Department of Computer Science, University of California, Irvine. His interests focus on bioinformatics, health informatics and artificial intelligence-aided healthcare.



Jiao Du was born in Chongqing, P.R.China, in 1988. She received M.S. and Ph. D degree from Chongqing University of Posts and Telecommunications, Chongqing, P.R.China in 2013 and 2017, respectively. Currently, she is a lecturer with the school the School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China. Her research interests include pattern recognition and image fusion.



Hengliang Tan received his B.E. degree from Foshan University, Foshan, China, in 2006 and his M.E. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2011 and 2016, respectively. He joined the School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China in 2016. His current research interests include machine learning, pattern recognition and manifold learning.



SECURE AND PRIVACY-AWARE DATA COLLECTION ARCHITECTURE APPROACH IN FOG NODE BASED DISTRIBUTED IOT ENVIRONMENT

Moussa WITTI and Prof. Dimitri KONSTANTAS

Information Science Institute University of Geneva
Route de Drize 7, 1227 Carouge, Switzerland

ABSTRACT

In the era of Internet of Things, data are collected from heterogeneous wireless protocols such as ZigBee, WiFi, RFID, Bluetooth, sub-GHz, Z-Wave, 2G / 3G / 4G from smart sensors to fog and cloud platform. However, the collected data may contains sensitive information, which the owner does not want to be disclosure. Because of IOT architecture based on heterogeneous technologies, ensuring privacy and maintaining security are difficult. How to protect data and preserve privacy over network during end-to- end or hop-to-hop communication? In this paper, we propose an architecture approach for secure and privacy-aware data collection in Fog Node Based Distributed IOT environment.

KEYWORDS

Internet of Thing, privacy, security, data collection, fog

1. INTRODUCTION

The growth of smart devices communicated together or via a distributed platform has enabled data collection from sensors to fog/cloud in IOT environment. Each sensor is able to transmit collected data to a fog server. A multiple fog server sends all collect data to a cloud server, which performs data processing, analysis and monitoring. Data are transported thought a heterogeneous environment, stored, analysed and sometimes transformed during processing.

According to Gartner, by 2025, over 1 trillion smart sensors will be used around the world and more than half of these devices will concern latency sensitive applications [2][3] such as healthcare and smart city applications. Since fog computing has emerged to support latency sensitive applications interacting with edge and cloud platform. How should privacy be preserved, and how should security be ensured, while collecting data across the edge-fog-cloud environment? How should data be secured through life-cycle processes across the edge-fog-cloud? Furthermore, how should privacy-aware data collection be provided in a well-secured Fog Node-Based Distributed IOT environment?

In this paper, we aim to apply privacy and security requirements on some data identified and categorized as sensitive. Thus, we propose an architectural approach to secure and preserve privacy while data collection in IOT fog and cloud environment. The paper is structured as follows: after background and related work in Section 2, Section 3 focuses on privacy and security requirements in IOT-enabled platform, Section 4 presents our proposed approach,

Section 6 propose solution architecture, comparison analysis and Section 7 provides conclusions and gives direction for future work.

2. RELATED WORK AND BACKGROUND

Privacy and security issues are challenged and several security models for IoT have been designed. The rapid growth of IoT has extended Internet to any small smart devices in distributed environment [6] therefore has introduced a problem. As IoT environment is more heterogeneous, more complex [3] and maintaining security is very critical in distributed system as well as cloud and fog environment [4] [11] . Most research studies [6] [10] [19] [20] [21] [30] are focused on how to integrate security among application, perception and transport layers level for distributed or cloud environment such as IaaS (Infrastructure as a Service), SaaS (Software as Service), and PaaS (Platform as Service). To protect sensitive data a huge of privacy-preserving algorithms have been developed such as k-anonymity, l-diversity. The concept of k-anonymity has been introduced by L. Sweeney and P. Samarati [24] in order to preserve privacy. While l-diversity is a data anonymization technique based on generalization and suppression often with a loss of the quality of the information. L-diversity is defined as extension of the k-anonymity [15]. Another algorithm ‘t-closeness’ [21] has been developed to anonymize data [15] [25] This technique is an extension of l-diversity and designed to preserve the confidentiality of sensitive data while reducing the granularity of data representation.

Several framework has been designed to maintain security along to end-to-end communication in IoT-based solu-tions. Cisco has proposed IoT/M2M Security Framework to protect data confidentiality and provide role-based security mechanisms. Other such as Icon Labs’ Floodgate Security Framework provides cyber security standards for Industrial Automation and management Systems (IACS) according to ISA/IEC 6244 standard.

2.1 Privacy and confidentiality

There is no universal definition of privacy because it differs according to the economic, societal, religious and cultural characteristics of a given population [8]. This means that privacy depends on our preferences what we want to share as information without disclosing personal matters. Many factors affect what people consider private. Many factors influence what a person may consider private. It depends mainly on the culture and the societal context. It also depends on a given situation according to which the same information considered as private differently [13]. Other researchers like American law professor Alan Westin have defined three levels of privacy norms: political, socio-cultural and personal level [19] [23]. Other searcher as Daniel Solove has tried to classify the elements of privacy [26] according to six categories such as:

- the right to be left alone,
- a secret access
- the control of personal information
- identity of the person
- Privacy.

2.2. Privacy policy and law regulations

Privacy rules implementation dependent on the context of the society and country laws:

- European Union has implemented General Data Protection Regulation and Data Protection Directive to protect privacy. The article 8 of European Convention on Human Rights (ECHR)

protect the individual and family right and privacy.

- United States have adopted three main federal laws which are Children's Online Privacy Protection Act (COPPA) to protected children under 13 age, Gramm-Leach-Bliley Act for privacy in financial institutions, Health Insurance Portability and Accountability Act for insurance companies use.
- Canada federal government provide Personal Information Protection and Electronic Documents Act (PIPEDA) to preserve privacy in data collection and electronic exchange.
- India government adopted by the end of 2000, The Information Technology Act 2000 improved in 2008 and in 2011 to integrate security practices and procedures to protect personal data or sensitive information.

2.3. Privacy concerns in Data Collection

Nowadays, a huge amount of data is collected from smart sensors and sent to fog and cloud processing system. IOT-enabled platforms must implement security and data protection rules following existing laws and regulations. The principle of privacy must be guaranteed. Sensitive data must be protected from attack and unauthorized access.

Because of the use of the Internet, IOTs inherit the same vulnerabilities as any computer device. How to preserve privacy and ensure security. Indeed IOTs are all potential victims of cyberattacks. An attack on a connected object can cause considerable damage to an IoT-enabled fog and cloud computing platform. From one point after a connection to a device communicating with the others, it possible to an attacker to can access the entire IoT-enabled platform. This creates a serious vulnerability and any confidential information on the network can be viewed from any connected device.

2.4. IOT main threats

Any IOT-enabled platform may experience the following types of attacks such as:

- DDOS (Distributed Denial of Service): massive attack on a network or a connected object in order to cause unavailability of the service or the server.
- Thingbot: multiple attacks from a network of large-scale cyber-attacks to take remote control of a connected object and spread malicious programs or access confidential data on an IOT platform.
- Man-in-the-Middle: interception of messages between two users by a malicious cyber-attacker with modification of the original message. Many MIM attacks on the IOT platform have been reported in smart Home and in the automotive era with connected object.

3. PRIVACY AND SECURITY REQUIREMENT IN IOT-ENABLED PLATFORM

Due to IOT architecture and its ubiquitous Internet connection [4], [12], [27], [32] maintaining security in IOT platform becomes more difficult.

3.1. IOT platform security requirements

Security requirements in IOT based architecture should be implemented along multi-layers [3][30]:

- Securing the perception layer:
- Securing the transport channel at the network layer using Transport Layer Security

(TLS), which is an encryption protocol to protect messages on the network, and provide secure channel to ensure privacy and data security.

- Securing data, files systems, and business applications at application layer

Yang et al. [30] has proposed a set of trust enhancing in IOT platform based on Key Exchange Management. Others as Bawany et al [3] have proposed an IOT security framework to prevent DDOS.

Thus, according to Yang et al. [30] and Bawany et al [3], an IOT-enabled platform should implement:

- IAM (Identity and Access Management),
- AAA (Authentication Authorization Accounting),
- K.E.M (Key Exchange and Management) for trust, and data integrity, confidentiality, availability, cryptography,
- I.A.A (Identification Authentication and Authorization)
- Devices resilience
- Trust: smart device trust and data trust
- Privacy: Data privacy, anonymity, unlinkability, unobservability, pseudonymity
- Network Security: TSL protocol
- Privacy-preserving policies: Data storage policy, location privacy, identity privacy, data processing and analytics privacy

Identity and Access Management (IAM) refers to users/groups identification and access to resources or applications. IoT-enabled platform IAM policies should implement identification mechanisms and role for users/groups to access a specified resource. Users belong to a group or multiple group with different roles. Multiple users may have the same role or privilege to access multiple resources.

IAM process is based on Authentication, Authorization, and Accounting (AAA) mechanism:

- User authentication refers to the process used to verify user's claims through login/password or smart card access, secret code, fingerprint scan, secure ID generated automatically by a program or smart key, etc.
- User authorization is mechanism performed to verify the user's access to resources or applications based on user's group, user's role or privileges.
- Accounting refers to the logging of the user's Authentication and Authorization mechanisms

Trust process in an IoT-enabled platform may be applied into data and devices level:

- Device trust refers to all mechanisms used to identify and recognize a component as trust and secure to communicate with other applications
- Data Trust defines the entire process to ensure that the data has never been altered during transport on the network. Data should be identical from the origin

Network security is built around three main objectives that are:

- Data Confidentiality: protecting data from unauthorized users
- Data integrity: ensuring data reliable and identical as from the origin

- Data availability: ensuring data available on the network for the right users when it is requested

3.2. Data security

We have organized data security approaches into four categories (see figure 1):

- Integrity and confidentiality of sensitive data to prevent the risk of tampering or injection or falsification
- Authenticity: data received must be authentic at the origin
- Non-repudiation: transmitted data should not be unknown to the sender
- Availability: ensure data availability reliably.

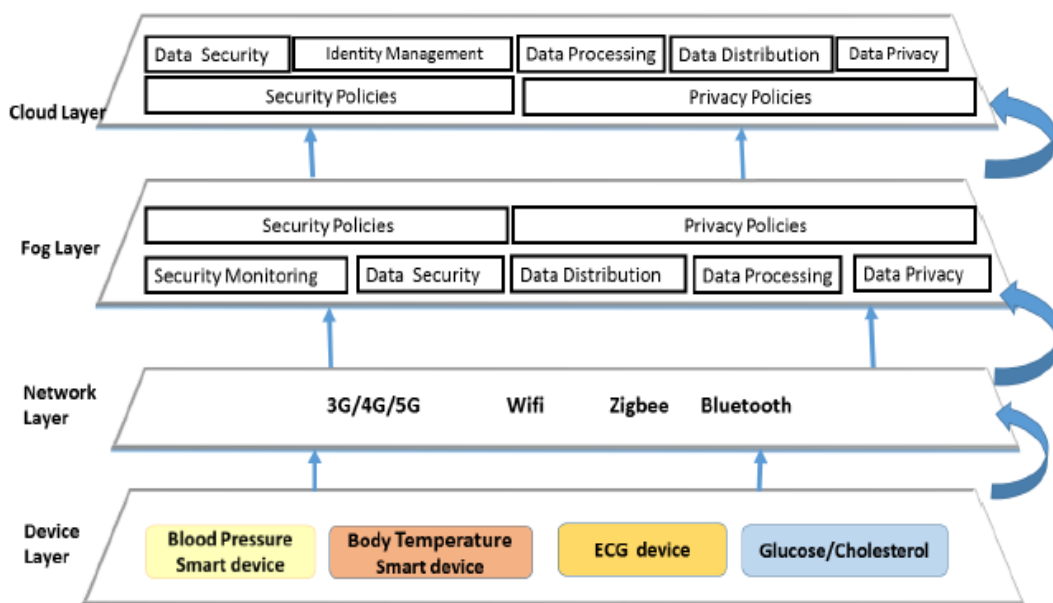


Figure 1: Data security and privacy management through IoT Layers

3.3. Data privacy

We have organized the privacy-preserving approaches into five categories (see figure 1):

- Privacy by cryptography
- Privacy by pseudonymization
- Privacy by anonymization
- Privacy by unlinkability,
- Privacy by unobservability

4. PROPOSED APPROACH

Data exchange in an IOT fog cloud environment may be secured in multiple manners and here we propose a bottom-up approach. We propose an architectural approach based on:

- 1- IOT devices identification/authentication/authorization process
- 2- Gateway and Wireless/Bluetooth access point control
- 3- Data protection while collecting using dynamic key and hash function on fog and cloud IOT environment

4.1. Device security

Most IOT devices in communication may be identified by an IP or media access control (MAC) address or by International Mobile Device Identity (IMEI). On the network, a malicious attacker can impersonate the IP address or MAC address to alter conveyed data or access to other resources.

In our approach, we propose life cycle Identity and Access Management (IAM) system in which each device must be identified by an ID on fog Nodes. A well-integrated IOT device security strategy must implement:

- Device identification system: in addition of IP and MAC address, any device must be authenticated by a specified ID with a role on the network
- Device identity lifecycle management system: device ID must change by the time to avoid spoofing in case of malicious attack. We define TTL - time to live.
- Device authentication and access control according security level and companies policies

Table 1. IOT devices security management.

IP Private	MAC	Device ID	Risk	Date	TTL
192.168.1.17	54:ff:7b:31:84:56	5266003410	Sensitive	2019-11-14.	15min
192.168.1.54	11:2a:7b:31:84:23	5556225620	Normal	2019-05-0	1 week
192.168.1.39	56:ff:7b:31:84:12	9963210263	Sensitive	2019-04-31	1 h
192.168.1.21	21:ff:7b:31:84:56	5200600500	Sensitive	2019-02-17	1h
192.168.1.65	63:ff:7b:31:84:56	3323822036	Normal	2019-06-09	1 week

4.2. IOT Access Point Control

Access point must be controlled according to the resource sensitivity. As devices are categorized according to data sensitivity, each devices has grant to a specified gateway to transmit data on the network. We propose a dynamic access control approach based on sensitive or non-sensitive data and associated risks. Only IOT device with a minimum of privileges can access to a control point or gateway.

4.3. Shared key online construction

In our approach, we propose to build a shared key online from conveyed data's elements such as token ID, user ID, data correlation ID (cf. table 1). All such elements will be placed in a matrix according to a specified order known both by client/server side. Thus, the secret key generation process will be reinforced.

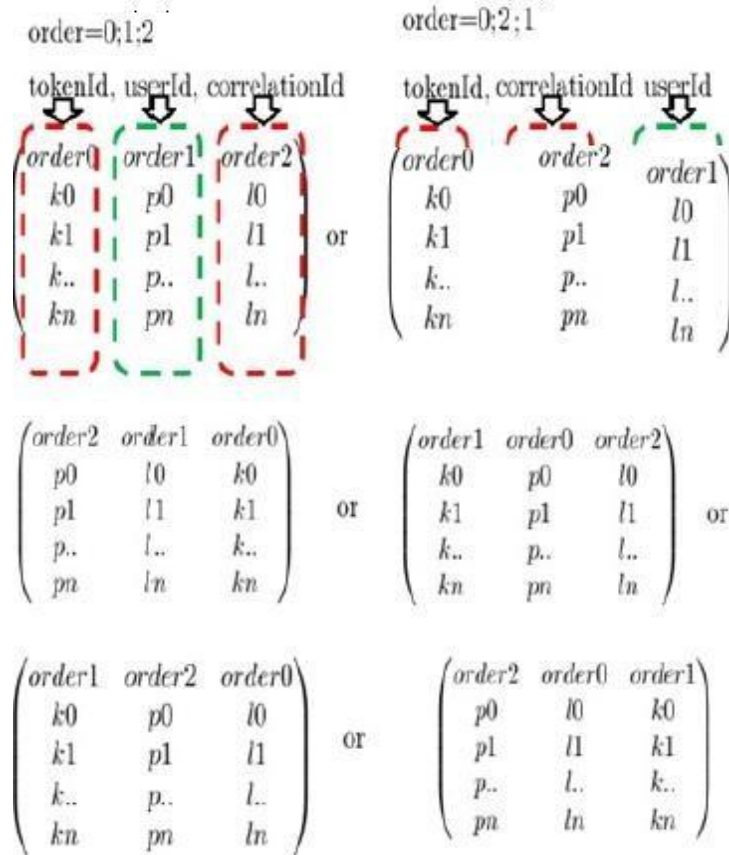


Figure 2: Secret Key Matrix Generation

The shared key is built online without complex calculations, which will not affect performance. The server knows all dealers identified by their ID. Collected data are identified by correlation ID on the server side in the cloud during processing analytic stage.

For each request, a new token Id I generated. The user ID is related to the dealer while correlation ID is depending on data collection program.

4.4. Data encryption/decryption

To ensure privacy, data should be encrypted before transmitting on the network. Dealers may use shared key to encrypt/decrypt data. We propose XOR operation to compute efficiently with all dealers. The proposed algorithm generate the shared key based on shared key generation which is based on Token ID, User ID and Correlation ID placed in different rank according to a specified order kown by the server and dealers (cf. figure 2).

Algorithm 1 Xoring Encrypt/Decrypt User Data

Input: *tokenId,userId, correlationId,order, data*
Output: *EncryptedData, or DecryptedData*

```

1: function ENCRYPTDECRYPT(userData)
2:   A=CreateMatrix(tokenId,userId, correlationId, order)
3:   K=GenerateKey(A)
4:   N ← length(data)
5:   for k ← i = 1 to N do
6:     result[i] = data.charAt(i) ⊕ key.charAt(i mod (key.length -1))
7:   end for
8:   return result
9: end function

```

5. PROPOSED ARCHITECTURE

As we defined an approach, we aims to propose secure and privacy-aware data collection solution architecture.

5.1. Our proposition

We propose a secure and privacy-preserving data collection architecture (cf. figure 3) based on:

- IOT identity management at device level: each device should send data first to a fog. All authentication, authorization, revocation and accountability process are managed on the fog node. Devices are authenticated by Id which is changing by the time, IP and MAC address. All unknown devices are revoked. Device recover process must be implemented for those which have an ID duration has expired.
- Privacy-aware data collection on the Fog Nodes: Shared key is generated according to token Id, and an order specified for each request, then data is encrypted using Xor operation before transmission on the network.
- Data decryption and processing on the cloud servers: using online the shared key based on token Id and an order in the response from fog node, the application on server side can decrypt data. Thus, privacy for all sensitive data are preserved.

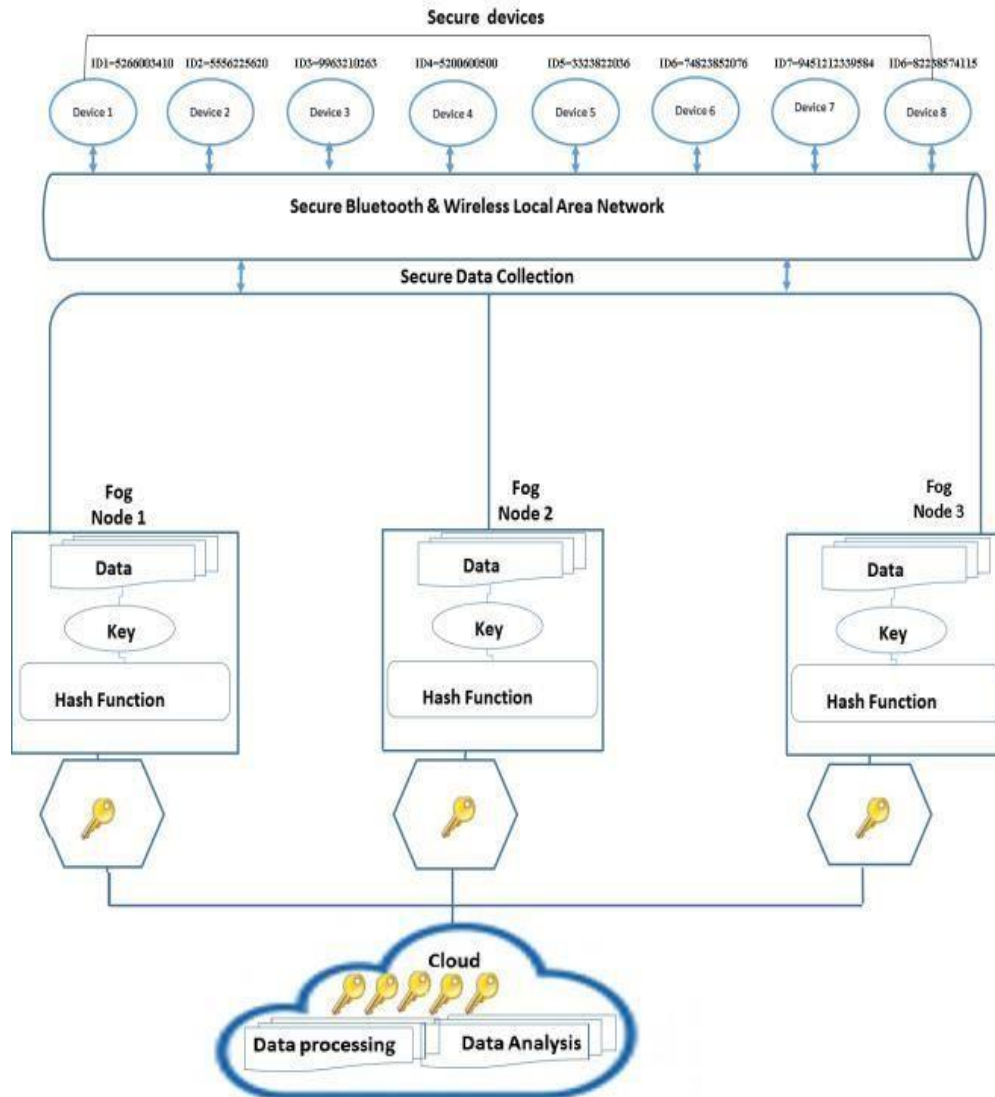


Figure 3: Proposed architecture

5.2. Prototyping And Implementation

We have implemented proposed solution using iFogSim [9] in Eclipse. To simplify our model, we assume that sensors exchange JSON format message. We implemented a fog platform to collect data from many devices. We created several broker, fog and edge devices using iFogSim and CloudSim toolkit.

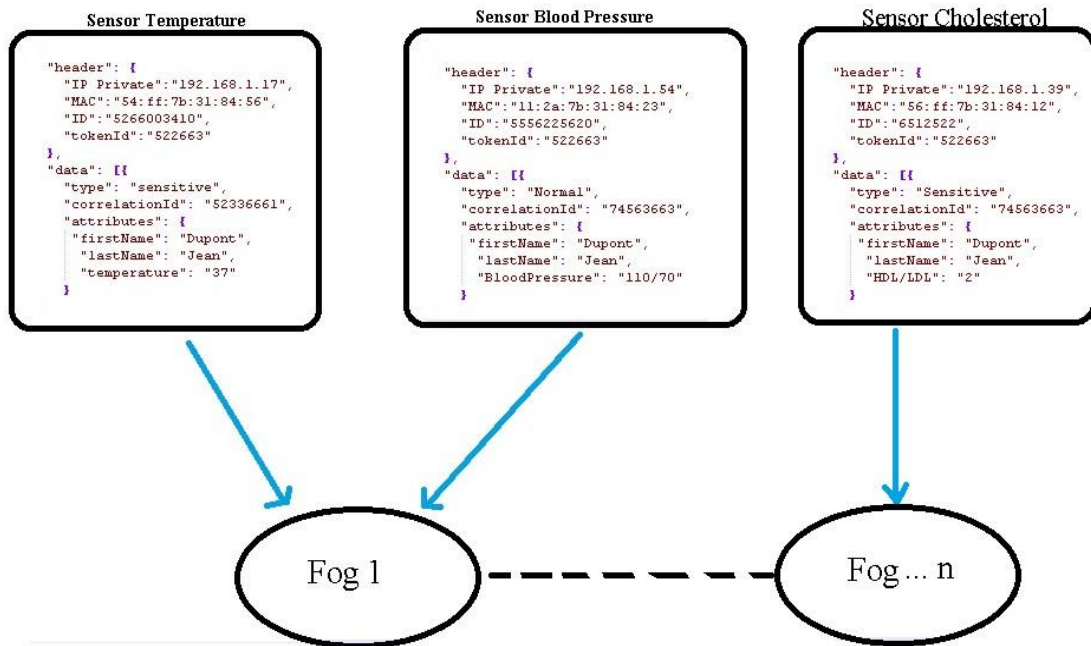


Figure 4: Prototype design

5.3. Data Encryption/Decryption Performance Analysis

We assess the performance of our proposed scheme (cf. subsection 4.4) comparing with AES algorithm. We can see that our scheme provide key generation from element conveyed in data and encrypting/decrypting process is more performant that AES as shown in the following picture.

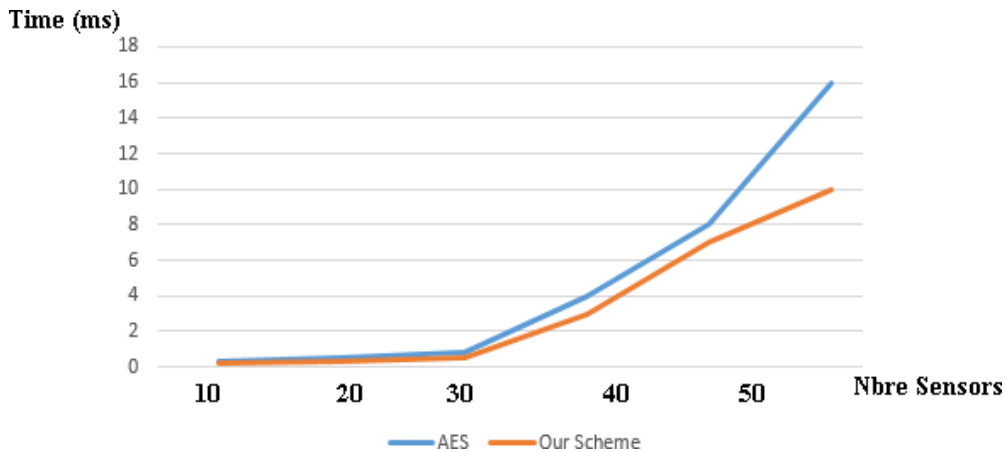


Figure 5: Our scheme vs AES performance analysis

6. DISCUSSION AND ANALYSIS

Maintaining security and privacy in IOT enabled platform becomes more difficult. Security and privacy requirements in IOT enabled architecture should be implemented along multi-layers:

- Securing device at the perception layer

- Securing the transport channel at the network layer
- Secure databases, files systems, and business applications at application layer

6.1. Discussion

The proposed architecture provide device security and preserve data integrity and confidentiality. All IOT devices are authenticated and their ID are management with possible revocation. A shared key generation mechanisms will encrypt data. Device traceability are guaranteed. A token ID is generated for each request. Device should be trusted by applying all mechanisms used to identify and recognize a component as trust and secure to communicate with other applications within IOT platform. Data should also be trusted using process to ensure that the data has never been altered during transport on the network. We should ensure that data should be identical from the source.

Device resilience refers the ability of a component to maintain service with alteration in the system environment while robustness refers to its resilience against attacks.

Thus, Data integrity and confidentiality are preserved. This solution prevent against Spoofing and Man in the middle attacks. Generated shared key based on token ID and an order will be different for each request.

A malicious attacker cannot access to device ID which is changing by the time (cf. section 4.1). The data encryption preserve data integrity and confidentiality. Data privacy are guaranteed in our architecture. Thus, the proposed model provides data privacy policies and device security and resiliencies against malicious attacks.

A malicious attacker cannot access to device ID that is changing by the time. The data encryption preserve data integrity and confidentiality and preserve data privacy comparing with other IOT architecture in the literature. Comparing with other IOT architectures, the proposed model provide data integrity, data confidentiality, data privacy policies, and device security and resiliencies against malicious attacks.

6.2. Comparison Analysis

We conducted a comparative analysis of the proposed architecture against other well-known framework (see Table 2) such as IoT@Work, BeTaa and OpenIoT. We remark that our and IoT@Work architecture are data privacy.

Table 2. Our proposition vs other IOT architecture.

Requirements	IoT@Work	BeTaas	OpenIoT	Our architecture
Device Security Management	---	---	---	+++
Data Integrity	+++	+++	+++	+++
Data Confidentiality	+++	+++	+++	+++
Network Security	+++	+++	+++	+++
Data Privacy	+++	--	--	+++
Resilience against attacks	--	--	--	+++
Data Encryption and Decryption Performance	--	--	--	+++

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new architecture based on shared key generation and data encryption on fog and cloud IOT enabled environment. Data integrity, data confidentiality and data privacy are preserved by data encryption mechanisms. Device are authenticated and authorized. IoT Device Identity Management process ensure traceability and revocability. The proposed architecture prevent malicious attacks such as Spoofing and Man in the middle attack. In the future work, we are planning to implement a real-life use case to assess security, data confidentiality preservation and performance in fog and cloud IOT-enabled distributed environment.

REFERENCES

- [1] Aaditya Jain, B. S. (2016, April). Internet of Things: Architecture, security goals, and challenges. *International Journal Innovative Research in Science & Engineering (IJIRSE)*, Vol.No2:Issue4
- [2] Alfaqih, T. M., & Al-Muhtadi, J. (2016). Internet of Things Security based on Devices Architecture. *International Journal of Computer Applications* (0975 – 8887).
- [3] Bawany, N.Z.; Shamsi, J.A.: Application layer DDoS attack defense framework for smart city using SDN. In: *Computer Science, Computer Engineering, and Social Media (CSCESM)* (2016)
- [4] Botta, A.; de Donato, W.; Persico, V.; Pescapé, A. Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems* 2016, 56, 684–700.
- [5] Chaqfeh, M.A.; Mohamed, N. Challenges in Middleware Solutions for the Internet of Things. In *Proceedings of the 2012 International Conference on Collaboration Technologies and Systems (CTS)*, Denver, CO, USA, 21–25 May 2012; pp. 21–26.
- [6] Elmaghraby, A. S., and M. M. Losavio. 2014. Cyber security challenges in Smart Cities: Safety, security and privacy. *Journal of Advanced Research* 5 (4): 491--497.
- [7] Frank D. McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, June 29-July 02, 2009, Providence, Rhode Island, USA
- [8] Fried, C.: Privacy. *The Yale Law Journal* Vol. 77, No. 3. (1968) p. 486. p. 475

- [9] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. 2016. iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments. arXiv preprint arXiv:1606.02007 (2016).
- [10] Hay M. , Kun Liu , G. Miklau , J. Pei , E. Terzi, Privacy-aware data management in information networks, Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, June 12-16, 2011, Athens, Greece
- [11] Jamil, D., and H. Zaki. 2011. CLOUD COMPUTING SECURITY. International Journal of Engineering Science and Technology 3 (4): 3478--3483. ProQuest SciTech Collection.
- [12] Lazarescu, M.T. Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. IEEE J. Emerg. Sel. Top. Circuits Syst. 2013, 3, 45–54.
- [13] Majtényi L.: Az információs szabadságok: adatvédelem és a közérdekű adatok nyilvánossága. Complex, Budapest, 2006. p. 211. Simon 2005. pp. 33-34.; Szabó 2005. p. 45.
- [14] Maram, B., Gnanasekar, J.M., Manogaran, G. et al. Service Oriented Computing and Applications March 2019, Volume 13, Issue 1, pp 3–15
- [15] Machanavajjhala A., Gehrke J., Kifer D., Venkatasubramanian M. l-diversity: Privacy beyond k-anonymity. 22nd International Conference on Data Engineering (ICDE'06), 24-24
- [16] Ndibanje, B., H.-J. Lee, and S.-G. Lee. 2014. Security Analysis and Improvements of Authentication and Access Control in the Internet of Things. Sensors (Basel, Switzerland) 14 (8): 14786--14805. Pmc.
- [17] Nissenbaum, H.: Protecting Privacy in an Information Age: the Problem of Privacy in Public. Law and Philosophy Vol. 17, No. 5-6. (1998) p. 581.
- [18] Ricardo Neisse, G. S. (2015). A Model-based Security Toolkit for the Internet of Things. ScienceDirect.
- [19] Roman R., Zhou J., and Lopez J., "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, vol. 57, no. 10, pp. 2266–2279, 2013.
- [20] Nakamura E.T., Ribeiro S.L., Privacy A, Security, Safety, Resilience and Reliability Focused Risk Assessment In a Health IoT System : Results from OCARIoT Project. IEEE Global Internet of Things Summit (GIoTS), June 2019.
- [21] Wang R, Zhu Y, Chen TS et al. Privacy-preserving algorithms for multiple sensitive attributes satisfying t-closeness. Journal of Computer Science and Technology, 2018, Volume 33, Number 6, Page 1231
- [22] Weber R. H., "Internet of things–new security and privacy challenges," Computer law & security review, vol. 26, no. 1, pp. 23–30, 2010.
- [23] Westin, A. F.: Social and political dimensions of privacy. Journal of Social Issues Vol 59, No. 2. (2003) pp. 431-434.
- [24] Samarati P. & Sweeney L., Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression. Technical Report SRI-CSL-98-04. Computer Science Laboratory, SRI International.1998.
- [25] Sicari S., Rizzardi A., Grieco L.A., Coen-Porisini A., Security, privacy and trust in Internet of Things, Computer Networks: The International Journal of Computer and Telecommunications Networking, v.76 n.C, p.146-164, January 2015.

- [26] Solove, Daniel J., «Conceptualizing Privacy» (2002) p. 1094.
- [27] Xiao L, H. B. (2010). A knowledgeable security model for distributed health information systems. *Computers & Security*, (pp. 331-349).
- [28] Xi-Jun Lin , Lin Sun , Haipeng Qu, Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications, *Computers and Security*, v.48 n.C, p.142-149, February 2015.
- [29] Xin Ma, Q. H. (2010). Study on the Applications of Internet of Things in the Field of Public Safety. *China Safety Science Journal*, 20(007):170-176.
- [30] Yang X., Z. L. (2012). "A multi-layer security model for internet of things," in *Internet of Things*. Springer, 388-393.
- [31] Yunjung Lee, Y. P. (2015). "Security Threats Analysis and Considerations for Internet of Things". 2015 8th International Conference on Security Technology (SecTech), (pp. vol. 00, no., pp. 28-30).
- [32] Zhang W., B. Q. (2013). Security Architecture of the Internet of Things Oriented to Perceptual Layer. in *International Journal on Computer, Consumer and Control (IJ3C)*, Volume 2, No.2.
- [33] Zhiqiang Yang, S. Z. (2005). Anonymity-preserving data collection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD '05)*. ACM, New York, NY, USA, (pp. 334-343).
- [34] Ziegeldorf J.H., Morchon O.G., Wehrle K. Privacy in the Internet of Things: Threats and challenges *Security and Communication Networks*, 7 (12) (2014), pp. 2728-2742

AUTHORS

Moussa WITTI is a consulting engineer and IT architect in the R&D. He is advising bank and insurance firms in content and data management. He has more than 13 years of IT application development and deployment experience. He has obtained an MBA from Toulouse Business School and master Research in Computer Science from university of Franche-Comté in Besançon (FRANCE).



Dimitri Konstantas is Professor at the University of Geneva (CH) and director of the . He has been active since 1987 in research in the areas of Object Oriented systems, agent technologies, and mobile health systems, with numerous publications in international conferences and journals. His current interests are Mobile Services and Applications with special focus in the well-being services for elderly and information security. Prof. Konstantas has a long participation in European research and industrial projects and is consultant and expert to several European companies and governments.



QUALITY MODEL TO THE ADAPTIVE GUIDANCE

Hamid Khemissa¹ and Mourad Oussala²

¹Computer Systems Laboratory, Faculty of Electronics and Informatics,
Computer Science Institute,
USTHB: University of Science and Technology Houari Boumediene, Algiers;
Algeria.

²Laboratoire des Sciences du Numérique de Nantes (LS2N), Faculty of
sciences, Nantes University, France.

ABSTRACT

The need for adaptive guidance systems is now recognized for all software development processes. The new needs generated by the mobility context for software development led these guidance systems to both quality and ability adaptation to the possible variations of the development context. This paper deals with the adaptive guidance quality to satisfy the developer's guidance needs. We propose a quality model to the adaptive guidance. This model offers a more detailed description of the quality factors of guidance service adaptation. This description aims to assess the quality level of each guidance adaptation factor and therefore the evaluation of the adaptive quality guidance services.

KEYWORDS

Quality model, Guidance System Quality, Adaptive Guidance, Plasticity.

1. INTRODUCTION

Due to technological progress, the developer is considered nowadays as a mobile actor operating in various development context using variable platforms. This trend seems interesting, however, it only poses a problem in the ability and quality adaptation to the possible variations of the development context (Garcia and Pacheco, 2009; Kirk *et al.*, 2009).

For this, it is necessary to assist developers and ensure the plasticity of the adaptive guidance systems (Calvary *et al.*, 2002; Coutaz, 2010; Khemissa *et al.*, 2012; Khemissa *et al.*, 2014) with their ability to adapt to the current development context, defined by the triplet (material platform, developer profile, activity context), in respect of their usefulness. Usefulness refers rigorously to quality services offered to developers. It refers to the ability of a guidance system that allows the developer to reach his objective preserving consistency and product quality in software development.

Finally, a quality guidance system is a system capable to satisfy the developer's guidance needs. Therefore, the system quality is estimated as a set of protocols and principles to be applied during the use of the guidance system to meet those needs.

In a first stage, our work is rather focused on the study and synthesis of the limits of the existing

software process modeling environments (Calvary *et al*, 2002; Coutaz, 2010; Khemissa *et al*, 2012). Taking into account specific factors for an adaptive guidance, we have classified these limits through retained factors describing explicitly the basic concepts linked to the adaptive guidance aspect (Khemissa *et al*, 2012; Khemissa *et al*, 2014). To realize the effectiveness of plasticity concept of the guidance system supported by its adaptation ability to current development context, the selected guidance quality factors are defined by:

- ▶ **Guidance core:** The basic guidance is defined as global orientations core regardless the profile of both the activity context and the actor.
- ▶ **Developer profile oriented guidance:** the guidance orientations are defined on the basis that the human actor, regardless his profile, has a central role in the progress of the development process.
- ▶ **Guidance to activity context:** The selection of the appropriate type of guidance is more often not adapted nor suitable to a current activity context.
- ▶ **Guidance types:** the selection of guidance types remains defined in a manual and intuitive way. It depends on the project manager experience and informal personality.
- ▶ **Plasticity of guidance:** the guidance functions are defined and offered on the basis that the human actor always operates on a uniform development context. It should be noted that the plasticity factor is not invoked by the existing software process environments and meta-models. It is typical to our modeling approach of the adaptive guidance.

Based on the specific factors for adaptive guidance, the environments and meta-models considered for a comparative study are: SPEM (OMG, Inc, 2008) and APEL (Estublier *et al*, 2003) considered as the most representative in the software process modeling, RHODES (Coulette *et al*, 2000; Tran *et al*, 2003) that uses basic concepts closest to those introduced by the proposed approach.

According to SPEM, the guidance is a describable element which provides additional information to define the modeling describable elements. However, the proposed guidance is not suitable to the profile components in the development context. The guidance is rather defined in an intuitive way.

ADELE/APEL is designed on reactive database. It proposes a global assistance of proscriptive type without considering the development context profile and automates part of the development process using triggers.

RHODES/PBOOL+ uses an explicit description of a development process. The activities are associated to a guidance system with various scenarios of possible realization.

We noticed well the global guidance aspect and limits for each meta-model. However, the current tendency is that developers would like to have a guidance quality intervention adapted to specific needs according to the characteristics of the current development context.

In this context, we have proposed an approach to define adaptive guidance modeling in software process. It has been described through a meta-model denoted PGM (Plasticity of Guidance Meta model) based on the concepts of development context's profile (Khemissa *et al*, 2012; Khemissa *et al*, 2014). This approach is defined in a Y description of the adaptive guidance. This description will focus on the three considered dimensions defined by the development context, the adaptation form and the provided service.

Each dimension considers several factors to deduce automatically the appropriate guidance service to be provided to developers according to the current context. The description of the first dimension offers an orientation base of the guidance regarding the profile of both the developer and the activity context. The second dimension defines guidance types to consider explicitly in a specific situation of the development context. The third dimension describes the possible adaptation form of guidance core. Finally, the plasticity of guidance is explained by the functional interrelation between these three dimensions. This approach is described schematically as follows:

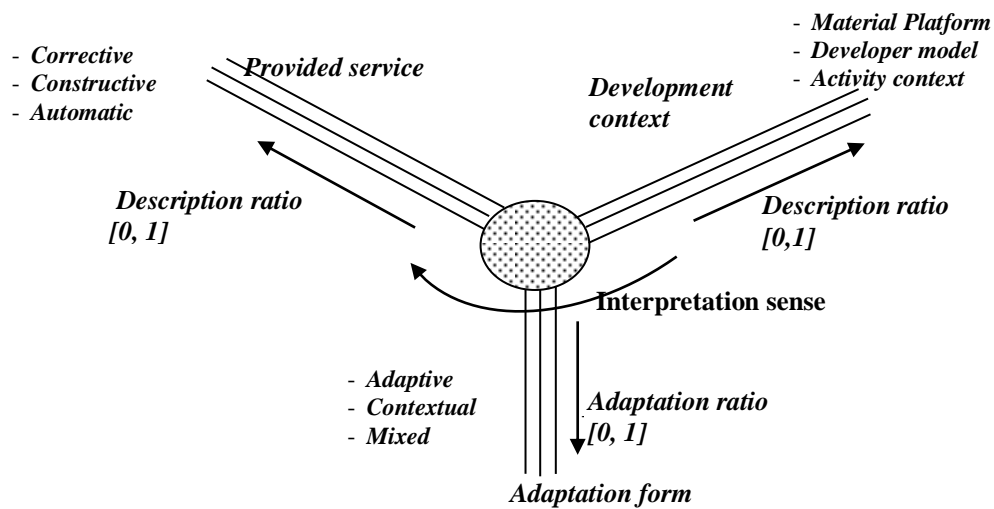


Figure 1. Adaptive guidance in Y description

In this perspective, and with a continuity spirit, we propose in this paper a quality model to the adaptive guidance. This model offers a refined description of the quality factors of the guidance service plasticity and adaptation. This description aims to assess the quality level of each guidance adaptation factor and therefore the evaluation of the adaptive guidance service quality.

2. QUALITY MODEL FOR THE ADAPTIVE GUIDANCE

In general, measuring the quality of a guidance system consists then in determining its appropriateness relatively to the guidance adaptation of the functional point of view. Getting a quality measure provides a clear picture of the guidance system and determines its behavior over time in terms of its adaptability to the development context. To have a complete clear image of the guidance system quality, we should define a quality model (Mordal-Manet *et al*, 2011; Mordal-Manet *et al*, 2013).

The most currently known models are hierarchical models that identify the quality principles, starting with the overall requirements and the most general principles to reach the technical criteria and associated metrics. These quality models offer both an overview of the system quality as well as a detailed view according to the considered point of view. They also allow to go from a detailed view to a global view and vice versa (Mordal-Manet *et al*, 2013).

Inspired by the quality model Mc Call (McCall *et al*, 1976), ISO 9126 model (ISO/IEC. Iso/iec 9126-3, 2003), ISO 25010 model (ISO/IEC. Iso/iec 25010, 2011) and Square norm (ISO/IEC. Iso/iec 25000, 2014; Balmas *et al*, 2010) recognized as international standard norms for assessing software quality. The development of our quality model is defined as a four-level model called: point of view-factors-criteria-metrics. It is identified through three points of view associated to the development context representing a global vision of quality.

Each of these point of views is described through five quality factors representing a quality external view. These factors are characterized by eighteen criteria that represent the quality internal view. These criteria are matched with the metrics that evaluate each criterion.

This metric is made on the basis of a process to evaluate quantitatively and semantically each criterion and therefore each of the quality factors and viewpoint in order to deduce every time the adaptive guidance quality according to the considered viewpoint.

The design pattern of the proposed quality model defined by four hierarchical description levels is schematically represented by the following figure.

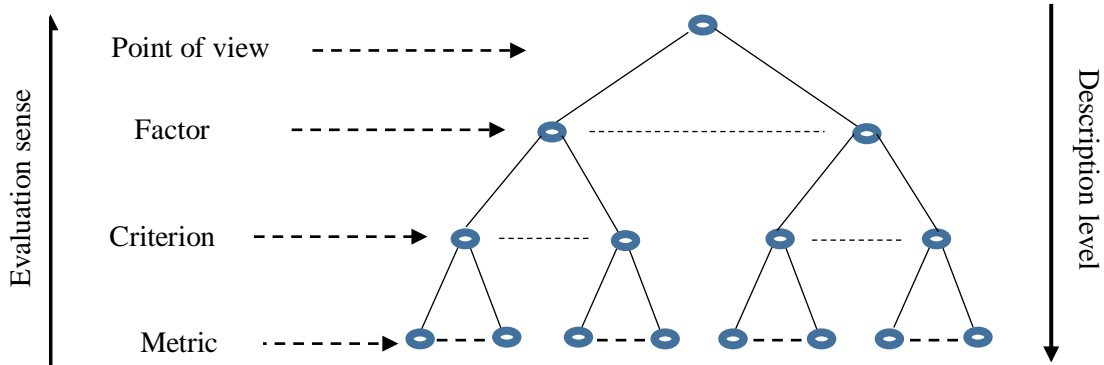


Figure 2. The design pattern of the quality model.

It can be instantiated to describe a specific quality model to a particular domain by describing the set of data related to each level relative to the considered point of view.

In our case, we consider three points of view related to the first dimension considered in our approach ‘‘PGM’’ namely the development context. The quality model for the developer point of view is described by the following diagram.

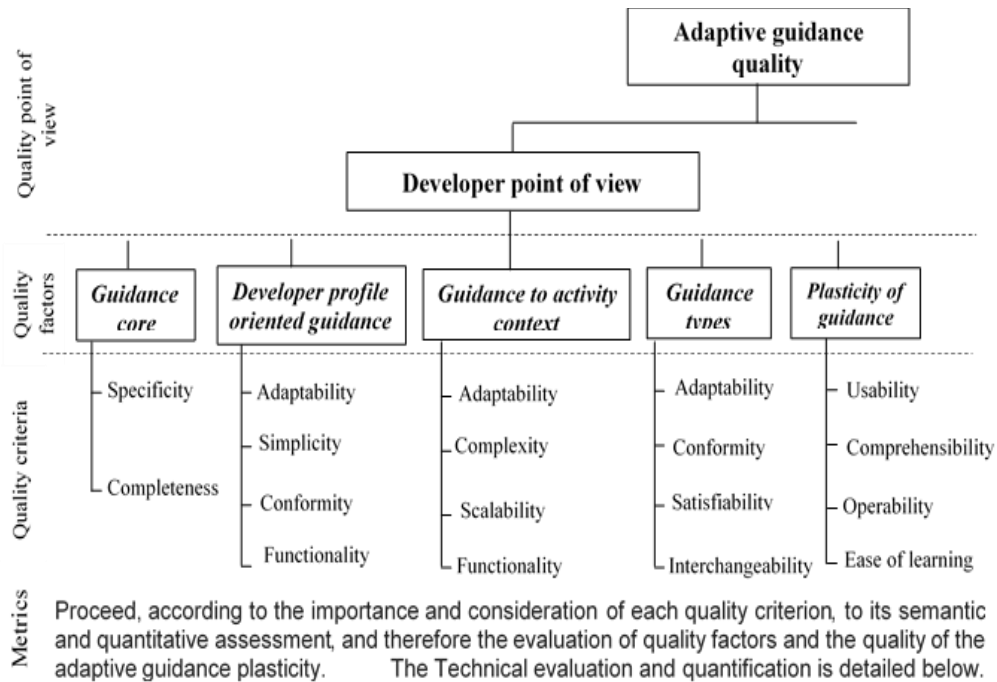


Figure 3. Quality model of the guidance plasticity

The detailed description of our quality model offers a more refined description of our quality factors through the specificity of the corresponding quality criteria. Each of the selected quality factors is described through a set of criteria as follows:

❖ **Guidance core:** this factor is decomposed and evaluated on the basis of the following two criteria:

- **Adaptability:** the degree from which the offered guidance can accommodate with specific situations of activity context.
- **Completeness:** the degree from which the guidance system provides coverage of the whole life cycle of a software process.

❖ **Developer profile oriented guidance:** this factor is discussed relatively to the following four criteria:

- **Adaptability:** the degree from which a guidance system considers, on the basis of the developer profile, all the elements relating to the three dimensions of adaptive guidance.
- **Simplicity:** the degree from which a guidance system can be used to achieve the goals identified by the performer efficiently and satisfiability in a specified activity context.
- **Conformity:** the degree from which a guidance system serves exactly the developer profile needs in a particular activity context.
- **Functionality:** the degree from which a typical system offers guidance services to support the developer needs in specific conditions.

❖ **Guidance to activity context:** this factor is evaluated in relation to the following four criteria:

- **Adaptability:** the degree from which a guidance system considers, based on the current activity context, all elements relating to the three adaptive guidance dimensions.
- **Complexity:** the degree from which a guidance system processes and addresses the needs of the current activity context.
- **Scalability:** the degree from which a guidance system provides the most appropriate behaviour to support the needs of the current context evolution.
- **Functionality:** the degree from which a guidance system offers typical guidance services to address the needs of the current activity context in specific conditions.

❖ **Guidance types :** this factor is appreciated on the basis of the following criteria:

- **Adaptability:** the degree from which a guidance system considers, based on the guidance type, all elements relating to the three adaptive guidance dimensions.
- **Conformity:** the degree from which a guidance system serves exactly the developer needs in a particular activity context.
- **Satisfiability:** the degree from which the offered guidance type ensures the developer needs in the current activity context.

- **Interchangeability:** the degree from which a guidance system supports the consideration of the various guidance services to address the developer needs in the given activity context.

❖ **Plasticity of guidance:** this critical factor targets the degree of a guidance plasticity through the following criteria:

- **Usability:** the degree from which a guidance system can be used on different activity contexts allowing to achieve the goals identified by the performer efficiently and satisfiability.
- **Comprehensibility:** the degree from which a system provides well-structured guidance services to support the developer needs in a given situation.
- **Operability:** the degree from which a system provides a mechanism allowing, at any time, the developer to call the guidance services related to the current activity context.
- **Ease of learning:** the degree from which a guidance system provides a support and learning service to support the concept of adaptive guidance.

3. QUALITATIVE EVALUATION PROCESS OF THE ADAPTIVE GUIDANCE

A metric is defined as a quantitative scale and a method which can be employed to determine the value taken by a property or a guidance system criterion.

The evaluation of the adaptive guidance quality is deduced by a practical process at four decomposition levels described by quality metric, quality criterion, quality factor and quality point of view.

This process is defined on the basis of the whole metrics, criteria, quality factors and points of view considered in a given software process environment.

3.1. First level: quality metric

The metric method for the quantitative assessment of each criterion is defined on the basis of the three dimensions considered in the Y description of the adaptive guidance. This metric observes each dimension impact with the involvement or non-consideration of its elements.

This method uses a binary process to note the involvement of each element of the adaptive guidance dimension. The value one " 1 " is associated to each element involved in the criterion evaluation. Since each dimension is defined through three basic components, the expression for evaluating the impact of each dimension is given by:

$$\text{Value Involvement Dimension} = \text{VID} = \Sigma (\text{Value element}_i) / 3 \quad \text{with } i = 1 \text{ to } 3.$$

Finally, the deduction of the quality criterion estimation is based on a mathematical expression combining all elements that define the implication of the three considered dimensions. For the quantitative evaluation, we use the formula of the simple average defined as:

$$\text{Criterion Measure} = (\text{VI context} + \text{VI service} + \text{VI form}) / 3.$$

With:

VI context: involvement value of the development context.

VI service: involvement value of the guidance service.

VI form : involvement value of the adaptation form.

Application example: evaluation of the 'Adaptability' criterion for the RHODES environment. The estimation of the 'Adaptability' criterion for the RHODES environment is generated on the basis of the consideration degree of each dimension. Assessing each dimension is made on the basis of its elements involvement. The application of the defined expressions gives us the following result:

$$\mathbf{VI\ context} = \Sigma (0 + 1 + 1) / 3 = 2/3.$$

$$\mathbf{VI\ service} = \Sigma (1 + 0 + 1) / 3 = 2/3.$$

$$\mathbf{VI\ form} = \Sigma (1 + 1 + 0) / 3 = 2/3.$$

$$\mathbf{Metric\ Adaptability} = (\mathbf{VI\ context} + \mathbf{VI\ service} + \mathbf{VI\ form}) / 3. \\ = (2/3 + 2/3 + 2/3) / 3 = \mathbf{0.66}.$$

3.2. Second level: quality criterion

Considering the evaluation technique of the proposed model by Boehm (Boehm *et al*, 2009) in the project management, the semantic quantification of each considered criteria is based on its contribution and impact on the adaptive guidance quality.

The semantic quantification process is done through three levels, described by high, medium or low contribution, applying the following rules:

<1/2: high order impact / = 1/2: middle order impact / >1/2: low order impact. Therefore, the numerical estimation of a criterion is done on the data interval [0, 1].

Application Example:

For the purpose of clarity, the following section provides a possible scenario to evaluate semantically the adaptive guidance quality criteria.

The numerical quantification of each criterion is deduced from the application of the defined methods describing the relationship between semantic evaluation and its numerical value.

3.3. Third level: quality factor

For each factor, the metric composition is carried at different levels from the measurements obtained at the criterion level.

A simple or weighted average often remains the most used way to compose metrics. The principle of the weighted average aims to promote the most influential criteria. The weight is applied to the criteria according to their influence degree.

In our case, the deduction of the estimated quality factor is based on a mathematical expression by combining the corresponding quality criteria. We use the weighted average formula.

Starting from a developer point of view and for a rigorous influence practice of each quality criterion, we associate the weighting 'Pi' according to the importance of each criterion. The Pi value varies over a range of 1 to n. n represents the largest number of considered criteria to describe a factor quality.

Finally, the computation of the quality factor value 'F_q' considers both the associated value of criteria quality and the corresponding weighting value for each criterion. This estimate is given by the following formula:

Adaptive guidance quality (developer's point of view) = $\sum F_i * P_{fi} / n$ with $i = 1$ to n .

With: **F_i**: quality factor estimation.

P_{fi}: associated weighting factor.

n: considered factors number.

Finally, the imbrication of the two preceding formulas, allows us to generate a combined expression for estimating the adaptive guidance quality. This combination is formalized by the following expression:

Adaptive guidance quality (developer's point of view) = $\sum (\sum C_i * P_i / n)_j * P_{fj} / m$

With: **i = 1 to n**, **n**: considered criteria number. **j = 1 to m**, **m**: associated factors number.

C_i: quality criterion estimation.

P_i: associated weighting criterion.

P_{fj}: associated weighting factor.

3.4. Adaptive guidance quality

It's also possible to measure the global quality of the adaptive guidance taking into account the three points of view. For this, we also use an average weighting with the necessity of affecting a weighting to the quality of each point of view. Therefore, we generate a layout combining the decomposition levels: criteria, factors and points of view. This combination is deduced by the following expression:

Global quality of the adaptive guidance = $\sum (\sum (\sum C_i * P_i / n)_j * P_{fj} / m) * P_{pv} / 3$

With: **i = 1 to n**, **n**: considered criteria number. **j = 1 to m**, **m**: associated factors number.

C_i: quality criterion estimation.

P_i: associated weighting criterion.

P_{fi}: associated weighting factor.

P_{pv}: associated weighting point of view.

4. PRACTICAL INTERPRETATION

The practical quality assessment for adaptive guidance is deduced by the quality metric based on the implication of each of the factors associated with the three dimensions considered by our "PGM" approach. This estimation is derived by two phases, the first phase is the semantic evaluation system describing the impact of the quality criteria on the guidance adaptation. The second phase involves the implementation of a digital process based on the formulas and methods defined by our approach through the four quality levels that is: quality metric, quality criteria, quality factor and quality point of view.

This interpretation will address the estimation of the adaptive guidance quality of RHODES environment (Coulette *et al*, 2000; Tran *et al*, 2003). It will focus on semantic evaluation of each criteria defined in relation to its strategy pattern description, its explicit description of the development process and the guidance system of the RHODES environment. The numerical estimation for each criterion is made by applying the defined methods and formulas defining the relationship between the semantic evaluation and numerical value.

Based on our study of the description and functioning of guidance developed by the RHODES environment, the semantic evaluation of each quality criterion is given as follows.

4.1. Quality criteria Evaluation

Based on the involvement principle of each factor associated with the three dimensions considered in our approach, the quantification process is carried by a digital process defined by the following rules:

- Semantic evaluation of "Medium" order is defined by the numerical estimation 1/2.
- Semantic assessments "High" and "Low" order are defined by inversely proportional quantifications, such as:

$$\text{Quantification (semantic evaluation = 1 - Quantification (semantic evaluation of high order))} \quad \text{Quantification (semantic evaluation of low order)}$$

The fact that the number of involved factors in each dimension is two, giving us an average of $(3 * (2/3) / 3)$. The application of this quantification process on the quality criteria of the RHODES environment is defined as follows.

4.2. Quality Factor evaluation

Considering a unique weighting criteria equals to 1, the evaluation of each quality factor on the environment RHODES is as follows:

$$F_q(\text{guidance core}) = (0.66+0.50) / 2 = 0.58.$$

$$F_q(\text{Developer profile oriented guidance}) = (0.66+0.66+0.66+0.50) / 4 = 0.62.$$

$$F_q(\text{guidance to activity context}) = (0.66+0.50+0.66+0.50) / 4 = 0.58.$$

$$F_q(\text{guidance types}) = (0.66+0.50+0.50+0.34) / 4 = 0.50.$$

$$F_q(\text{Plasticity of guidance}) = \text{-----}.$$

Besides, the factor "Plasticity of guidance" is not an invoked factor in RHODES, we note that the RHODES environment covers well the full range of quality factors.

4.3. Quality point of view evaluation

The guidance quality estimation considers both the factors quality value and the corresponding weighting value of each factor. With associated weighting equal to 1, the estimate formula is given as follows:

$$\text{Adaptive guidance quality (developer point of view)} = \sum F_i * P_{fi} / n \text{ with } i = 1 \text{ to } n \\ = (0.58+0.62+0.58+0.50)/4 = 0.58.$$

Finally, we can conclude that the adaptive guidance quality for RHODES environment through only the developer's point of view is estimated at 0.58 and therefore, it's pretty well taken into consideration.

5. CONCLUSIONS

Our main purpose in this article is to propose a quality model to the adaptive guidance system for software process modeling. This quality model is highlighted through a detailed description of the quality factors of guidance service adaptation. This description allows to evaluate the

quality level of each guidance adaptation factor in order to deduce the adaptive quality of guidance service.

The evaluation of the adaptive guidance quality is deduced by a practical process at four decomposition levels described by quality metric, quality criterion, quality factor and quality point of view. The developer point of view is described through five quality factors representing a quality external view. These factors are characterized by eighteen criteria representing the quality internal view. These criteria are matched with the metrics that evaluate each criterion. This metric is made on the basis of a process to evaluate quantitatively and semantically each criterion, and therefore each of the quality factors and point of view in order to deduce every time the adaptive guidance quality according to the considered point of view.

A perspective to this work concerns, at first, the necessity to estimate the productivity and cost due to the quality adaptation of guidance system. On another hand, we will also ensure the flexibility and adaptation of the metric system to the possible evolutions of the software process model.

REFERENCES

- [1] Garcia, I. and Pacheco, C. (2009): Toward Automated Support for Software Process Improvement Initiatives in Small and Medium Size Enterprises. Book chapter. Software Engineering Research, Management and Applications Volume 253, pp. 51–58. C_Springer-Verlag Berlin Heidelberg. ISBN: 978-3-642-05440-2.
- [2] Kirk, D.C., Macdonell, S.G., and Tempero, E. (2009): Modeling software processes - a focus on objectives, in Proceedings of the Onward. Conference. Orlando FL, USA, ACM Press, pp.941-948.
- [3] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., and Vanderdonckt, J. (2002): Plasticity of User Interfaces: A Revised Reference Framework. In: TAMODIA 2002.
- [4] Coutas, J. (2010): EICS '10. User interface plasticity: model driven engineering to the limit. Proceedings of the 2nd ACM SIGCHI symposium on engineering interactive computing systems. June 2010.
- [5] Khemissa, H., Ahmed-Nacer, M. and Oussalah, M. (2012): Adaptive Guidance based on Context Profile for Software Process Modeling. Information Technology and Computer Science, July 2012, 7, pp 50-60. Volume 4, number 7. DOI: 10.5815/ijitcs.2012.07.07.
- [6] Khemissa, H., Ahmed-Nacer, M. and Oussalah, M. (2014): Plasticity of a Guidance System for Software Process Modeling. First International Conference on Computer Science Information Technology (CoSIT), pp. 49–63, Bangalore, India. CS & IT-CSCP 2014. DOI: 10.5121/csit.2014.4905.
- [7] OMG. Inc. (2008): Software and System Process Engineering Meta-Model Specification version 2.0: Formal/2008-04-01.
- [8] Estublier, J., Villalobos, J., Tuyet lean H, Jamal-Sanlaville, S. AND Vega, G. (2003): An Approach and Framework for Extensible Process Support System. In Proceedings 9th European Workshop on Software Process Technology (EWSPT 2003), Helsinki, Finland, 2003-09-01.
- [9] Coulette B., Crégut X., Dong T.B.T. and Tran D.T., (2000): RHODES, a Process Component Centered Software Engineering Environment”, ICEIS2000, 2nd International Conference on Enterprise Information Systems, Stafford, pp 253-260, July 2000.

- [10] Tran Hanh Nhi, Coulette, B., Crégut, X., Thuy Dong Thi Bich, and Thu Tran Dan. (2003): Modélisation du méta-procédé RHODES avec SPEM. Dans : Recherche Informatique Vietnam-Francophone (RIVF'03), Hanoi, Vietnam.
- [11] Mordal-Manet, K., Laval, J. and Ducasse, S. (2011): Modèles de mesure de la qualité des logiciels, in Evolution et Rénovation des Systèmes Logiciels. Hermès 2011. <Hal 00639279>.
- [12] Mordal-Manet, K., Anquetil, N., Laval, J., Serebrenik, A., Vasilescu, B. and Ducasse, S. (2013) Software quality metrics aggregation in industry. In Journal of Software: Evolution and Process 25 (10) p. 1117—1135, 2013. DOI: 10.1002/smr.1558.
- [13] Mc Call, J., Richards, P. and Walters, G. (1976): Factors in Software Quality. NTIS Springfield.
- [14] ISO/IEC. Iso/iec 9126-3 software engineering -product quality- part 3: Internal metrics, 2003.
- [15] ISO/IEC. Iso/iec 25010-2011 software engineering -product quality- part 1: Quality model, 2011.
- [16] ISO/IEC. Iso/iec 25000-2014 software engineering-software product quality requirement and evaluation, 2014.
- [17] Balmas, F., Bellingard, F., Denier, S., Ducasse, S., Franchet, B., Laval, J., Mordal-Manet, K., and Vaillergues, P. (2010): The Squale Quality Model. INRIA-00533654, Version 1-Second Edition, 8 Nov 2010. <http://www.squale.org/quality-models-site/deliverables.html>
- [18] Boehm, B.W., Abts, C., Brown, A.W., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D.J., and Bert Steece, B. M. (2009): Software Cost Estimation with COCOMO II. Prentice Hall Edition, ISBN: 0137025769, 978013702576.

AUTHORS

Hamid Khemissa is a full associate professor at Computer Systems Department, Faculty of Electronics and Computer Science, USTHB University, Algiers. He is member of the software engineering team at computer system laboratory LSI, USTHB. His current research interests include Software Process Modeling and Software Modeling Assistance.

Mourad Chabane Oussalah is a full Professor of Computer Science at the University of Nantes and the chief of the software architecture modeling Team. His research concerns software architecture, object architecture and their evolution. He worked on several European projects (Esprit, Ist, ...). He is (and was) the leader of national project (France Telecom, Bouygues telecom, Aker-Yard-STX, ...). He earned a BS degree in Mathematics in 1983, and Habilitation thesis from the University of Montpellier in 1992.

A HYBRID MODEL FOR EVACUATION SIMULATION AND EFFICIENCY OPTIMIZATION IN LARGE COMPLEX BUILDINGS

Hao Yuan¹, Guo Yu², Yifan Ma¹, Jieneng Chen³, Xiongda Chen²

¹School of Software Engineering, Tongji University, China

²School of Mathematical Sciences, Tongji University, China

³School of Electronics and Information Engineering, Tongji University, China

ABSTRACT

The Cellular Automaton(CA) and Artificial Potential Field(APF) method, as well as other theories, are traditional to simulate the flow of people .A refine model of CA, combined with adapted Ant Colony model, as well as the APF is delivered to simulate the evacuation process in large buildings. An estimation of the total evacuation time within one floor in the Louvre is obtained by applying this model. The bottlenecks are identified alongside the evacuation routes. The applicability and flexibility of this model are proved.

KEYWORDS

Evacuation Model, Cellular Automaton, Artificial Potential Field, Ant Colony, The Louvre

1. INTRODUCTION

1.1. Background

Each year there are countless lamentable emergency evacuation failures being reported[1]. Especially, for the large areas with thousands of tourists, emergency evacuation plan plays an important role in ensuring the safety of the people inside, as it helps individuals leave the building as quickly and safely as possible. However, some previous studies [2,3] point out that with the lack of gates and complex structure, it would be rather difficult to evacuate all the people within the building.

In general, to help reduce the evacuation time and identify the bottlenecks during evacuation, it's of vital importance to obtain a evacuation simulation model, as it helps the administrators to get a better understanding of the building and also makes it easier for them to work out an evacuation plan.

1.2. Literature Review

In order to simulate the flow of people, a series of mathematical models are adopted, such as cellular automaton [4,5], which use the status change of a cellular to represent the movement of an individual. As the individuals in the cellular automaton move without intelligence and insight, while implementing the cellular automaton, some people seek to put the interaction between individuals into consideration and obtain the optimized route using ant colony algorithms [6,7]. After that, the cellular automation are modified with spatial refinement [8], which simulates the flow of people better. The cellular automaton is combined with artificial potential field [9] to generate a more precise route for individuals.

Unfortunately, most of these above don't depict the flow of people during the evacuation process precisely, as they haven't fully considered the intelligence of people and the interplay between individuals. Moreover, above research concentrates on evacuation in single simple room [4], which may go uncertain in large building with complex structure. Therefore, though there are many models existing, improvements are still needed.

1.3. Our Work

In order to carefully simulate the evacuation process in large complex buildings, a hybrid model integrating 3 sub models is developed, in which human movements at a finer granularity are depicted, and others' influence on the individual is considered.

The remaining part of this paper is arranged in the following order. How to develop this evacuation model from the three aspects of the simulation is given in Section 2. The second floor of the Louvre in France is utilized as an example for the model application in Section 3. Finally, some concluding remarks and the directions of subsequent research are provided in Section 4.

2. EVACUATION MODEL DEVELOPMENT

Intuitively, this hybrid model is divided into three sub models, namely Refined Cellular Automaton(RCA), Adapted Artificial Potential Field(AAPF), and Adapted Ant Colony Model(AAC), so that the flow of people during evacuation is simulated more realistically based on those three models.

2.1. Refined Cellular Automaton

RCA simulates the movement of individuals inside the building with complex structure. In this model, individuals will move in the direction where there is still place for them to move. As they don't have intelligence, individuals largely move randomly. According to the traditional CA[10], time and space are discretized, and the evacuation area is divided into discrete grids(cells). One cell represents the area taken up by an individual currently. However, such simulation is not fine enough in an evacuation scenario, as the speed of individuals are not carefully depicted, thus it is refined to get the RCA sub model.

2.1.1. Design of RCA

In RCA, in order to depict the speed of an individual and portray the evacuation process in more

detail, the original CA and the modifications are made in the model are listed as follows. The evacuation area are divided into more dense grids, and each grid corresponds to one sub-cell. Each cellular will cover several sub-cells (and in our model a cell occupies a 3×3 piece of sub-cells). When an individual moves, its movement will cover several sub-cells. The more sub-cells it covers per unit time, the faster it moves.

When a cell is determining its direction of movement, it will scan for the information contained in its Von Neumann type neighbor sub-cells[11]. After calculation, it will choose a most promising direction.

2.1.2. Mathematical Model of RCA

According to the idea above, RCA model is proposed. The individual's velocity is V as shown.

$$|V(i, j)| = \min\{v_0, \max\{n_{i+1, j}, n_{i-1, j}, n_{i, j+1}, n_{i, j-1}\}\} \quad (1)$$

$C(i, j)$ is the cell of row i , column j . In (1), $V(i, j)$ represents the velocity of the individual in $C(i, j)$. And v_0 is the velocity of this individual regardless of his surroundings. Then, $n_{a, b}$ is defined as (2).

$$n_{a, b} = \min\{d_{|(i, j)-(a_k, b_k)|}\} \quad (2)$$

In (2), (a_k, b_k) denotes all the cells where the individual has an overlap with $C(a, b)$. Moreover, $d_{|(i, j)-(a_k, b_k)|}$ represents the number of sub-cells between $C(i, j)$ and $C(a_k, b_k)$.

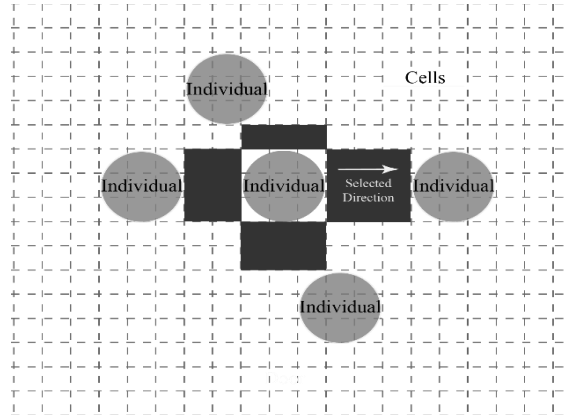


Figure 1: Individual's choice of moving direction

2.2. Adapted Artificial Potential Field Model

This sub model enables the individuals in our model to know the path to the doors when they are inside a room.

Artificial Potential Field (APF) serves as a method for local path planning[12,13], by assigning potential energy to the area and generating an APF. The object inside is able to get the route by setting the direction as the descending direction of the potential direction. Thus, it is integrated into this model so as help the individuals to get the route to the door.

However, APF appears to be weak when it comes to buildings with complex structures. In APF, the potential energy increases in line with the straight-line distance to the source. That means, when obstacles exist between the object and the source, the object cannot find a way to avoid them. Thus, the object in APF easily gets trapped when there are complex obstacles in the scene, or the room is U-shaped.

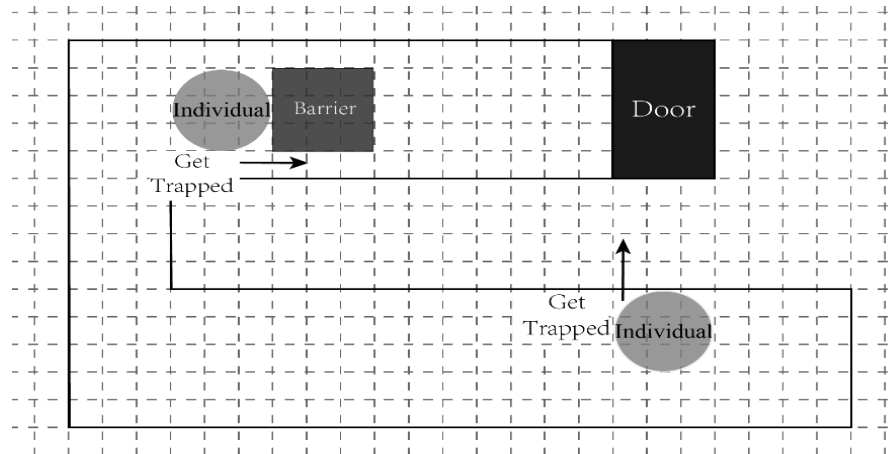


Figure 2: Individual gets trapped in APF under certain scenarios

To solve the problem, APF is modified as Adapted Artificial Potential Field(AAPF) in this hybrid model. The route is simulated along which the visitors will go for the doors spontaneously during evacuation if they are inside a room.

2.3. Design of AAPF

In AAPF, the distribution of potential energy within the area is modified so that the individuals inside can get to the door without being trapped by obstacles or the corners of the room.

The potential energy increases along the route in the room, not in line with the straight-line distance to the source. In this way, obstacles and corners inside the room will have no influence on the routes of the individuals as the distribution of potential energy has already bypassed them.

The doors are set as destinations, which possess the lowest potential energy. Thus, individuals inside a room will spontaneously move towards the doors, which is consistent with the reality.

AAPF and RCA are actually closely intertwined in an evacuation scenario. While RCA just depict the movement of the individuals, AAPF provides intelligence for the individuals, and enables them to 'see' the doors when they are inside a room and go for it instead of moving without an aim. Also, the potential energy is left in each cell, that is to say, by reading the potential energy left in his surrounding cells and getting the lowest one, the correct direction to the door is known by individual.

2.3.1. Mathematical Model of AAPF

The potential energy of cells and attraction of doors are normatively denoted as the following two functions. The potential energy of the i^{th} visitor when one is in the h^{th} room at the time t is

$$U_{ht}(i) = 12 \cdot \xi \cdot (\rho_{gt}(i))^2 \quad (3)$$

with ξ a constant. In (3), $\rho_{gt}(i)$ represents the Euclidean distance between the i^{th} visitor and the g^{th} door at time t where the g^{th} door means the door that the i^{th} visitor has selected. After that, the door's attraction is as follows.

$$F_{ht}(i) = -\nabla [U_{ht}(i)] = \xi \cdot (\rho_{gt}(i)) \quad (4)$$

(4) shows the door's attraction to the i^{th} visitor at time t with The potential energy of the i^{th} visitor when one is in the h^{th} room at the time t in (3) .

2.4. Adapted Ant Colony Model

The influence of people's own thoughts and minds of the simulation is included in this sub model. That is, individuals will be affected by others' choice when deciding which door they move towards.

The original Ant Colony Algorithm simulates the phenomenon in an ant colony, that the latter tends to follow the former along the pheromone left behind while they are moving. By tracking the pheromone whose concentration is inversely proportional to the length of the path, the ants are able to identify the optimal route[14]. To simulate the herd mentality of the individuals in the process of evacuation, it is integrated into this model so that the evacuation process is simulated more precisely.

In order to make it adapted to the situations where the structure of the building is complex, the phenomenon has to be left on the doors rather than along the route. Thus, the original Ant Colony model is modified to be Adapted Ant Colony model (AAC), so that it simulates how each individual's choice is influenced by others.

2.4.1. Design of AAC

In the adapted ant colony model, the modifications to the original Ant Colony Model are listed as follows.

The pheromone is only left on the door rather than alongside the specific path that an individual walk along. Until an individual arrives at one of the exits of the Louvre, the pheromone he spread begin to take effect. The numbers of doors on the route are to indicate the length of the route instead of using the actual path length.

That's to say, AAC simulates the process that every visitor chooses a specific door as his destination when he is inside a room.

2.4.2. Mathematical Model of AAC

In this paper, the probability that each visitor chooses a door is

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in D_k(t)} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}. \quad (5)$$

In (5), $\tau_{ij}(t)$ and $\eta_{ij}(t)$ respectively represent the concentration of pheromone at the j^{th} door and the inverse of the distance towards the j^{th} door at the time of t when the i^{th} visitor is at the i^{th} door. α and β respectively mean the weight index of $\tau_{ij}(t)$ and $\eta_{ij}(t)$. In addition, $D_k(t)$ refers to the doors that are available and passed by some successful visitors for the k^{th} visitor at time t .

As RCA considers the influence of the distance towards the doors inside a room, so weight β equals zero. As the concentration of the pheromone increases, the effect of the distance between D_i and D_j is ignored.(5) gives us the probability of the k^{th} visitor to choose the j^{th} door when he is at the i^{th} door at time t .

The way to get $\tau_{ij}(t)$ is as follows.

$$\tau_{ij}(t + \Delta t) = \rho \cdot \tau_{ij}(t) + \sum_{h \in S_k} \Delta \tau_{ij}^h \quad (6)$$

In (6), the time unit is set as Δt , where ρ represents the probability that pheromone is retained within Δt . Then $S_k(t)$ means those successful visitors that has passed $door_j$ through $door_i$ during Δt after time t with

$$\Delta \tau_{ij}^h = \frac{1}{L_h}. \quad (7)$$

L_h represents the number of the doors that the successful visitors have passed after $door_i$. $\tau_{ij}(t + \Delta t)$ is obtained by calculating $\tau_{ij}(t)$ using methods mentioned in (6).

3. MODEL APPLICATION AND ANALYSIS

Based on the hybrid model developed, it is applied to real-life scenarios to test the applicability. Here our model is applied to the Louvre to simulate an evacuation process with some real data got from Affluences[16]. After building the evacuation simulation, the evacuation process is analyzed and the bottlenecks are identified.

The Louvre has witnessed a series of shocking terror attacks taken place in France since 2012[17], and they have put French citizens as well as tourists at threat. Being one of the largest and the most popular art museum in France, the Louvre accepts an average of 15,000 visitors a day[18]. Thus, the Louvre needs a comprehensive and adaptable evacuation model, so that in the event of an emergency, its internal visitors can evacuate smoothly and minimize losses. That's why the Louvre is chosen to test our hybrid model.

3.1. Model Application

In order to apply the hybrid model to the Louvre precisely, data are gathered including the size[19] and the floor plan of the Louvre[20], and also get the real-time tourists' number from Affluences[12]. Using the data gathered, the 2nd floor model is constructed based on the floor plan. The halls into rooms are also divided according to the serial numbers in the floor plan. Numbers in Fig 3 are serial numbers of rooms.

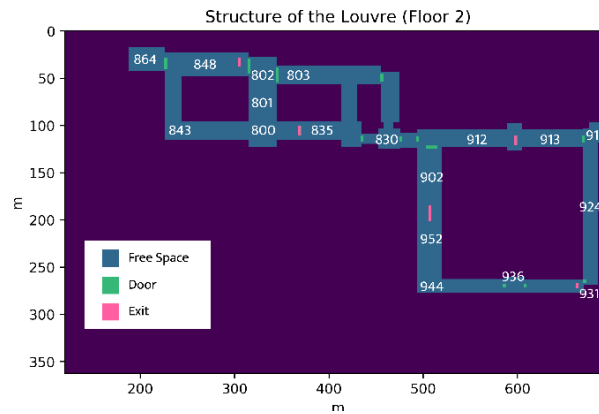


Figure 3: Structure of the Louvre (Floor 2)

3.1.1. Assumptions and Preparation

In order to simplify the course of modeling and draw some reasonable conclusions from the model, assumptions are as follows:

1. All the visitors will follow the guidance during evacuation.
2. When visitors arrive at the exits, they are able to leave the Louvre successfully. It is assumed that there won't be congestion outside the exits when emergencies happen.
3. The moving speed of the visitors are divided into two groups, and people within the same group move at a same speed. It's assumed that the visitors are divided into two groups: people who walk at normal speed, people who walk slower. The people who walk slower means those who have difficulty in moving, the elderly and the children for example.

Also, visitors are distributed to the five floors based on the floor area ratio of five floors. In this model, the number of visitors on the 2nd floor is 133.

3.1.2. Application of the Sub Models

Three sub models of this hybrid model are applied to the 2nd floor of the Louvre. Here how each of these sub models are applied to the scenario of evacuation is illustrated.

1. Refined Cell Automaton

After dividing the area into more dense grids, RCA has the ability to depict the speed difference between different kinds of individuals, such as the elderly and people at a younger age, by letting them cover a different number of cells at the same time.

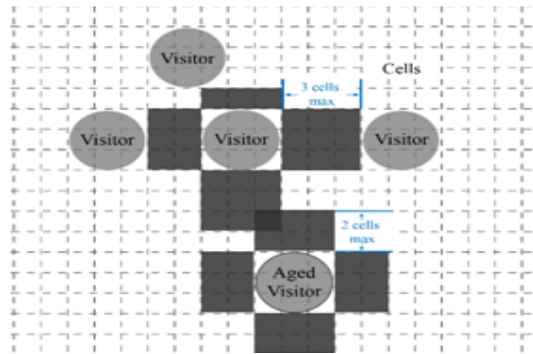


Figure 4: Speed difference between different kinds of visitors described using RCA

2. Adapted Artificial Potential Field

The route to the door for the individual inside a room is obtained using by the model of AAPF, as there are potential energy restored in each cells. The direction of an individual in which the cells have the lowest potential energy is chose by reading the potential energy in his surrounding cells and comparing them.

The choice of direction of a visitor in AAPF is shown in Figure 5. The gradient of color depicts the potential energy in the room. The darker the color, the greater the potential energy is here. Thus, the visitors inside will choose to move towards where the potential energy is relatively lower, that is, where the color is relatively lighter.

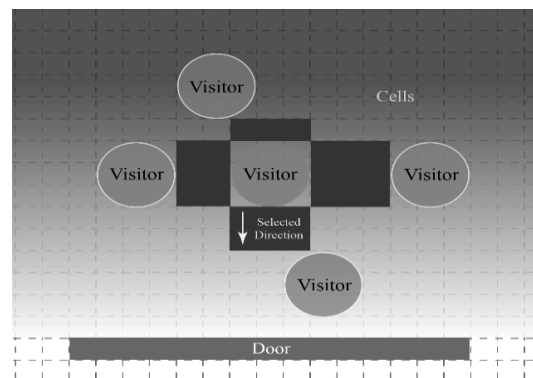


Figure 5: Choice of direction of visitor in AAPF

3. Adapted Ant Colony Model

AAC enables individuals to be affected by other individuals. When one individual gets to the exit and leave the building, pheromone will be left on the doors he passed. Combing with AAPF, the door on which the pheromone was left will be able to create an AAPF within the room, enabling the individuals inside to be guided to this door. Through such method, the individuals behind will be able to know through which door they can get to the exit.

Figure 6 shows the pheromone a visitor left after he arrived at the exit successfully. The arrows indicate the marks he left on the doors in his forward route.

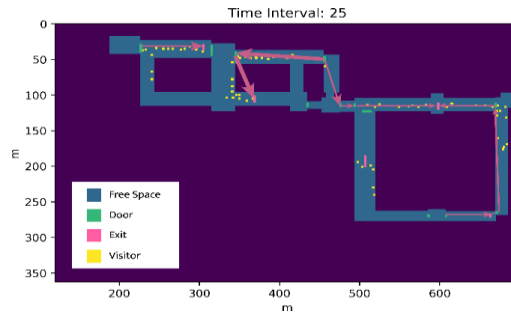


Figure 6: Pheromone a visitor left after he arrived at the exit

3.2. Analysis of the Simulation Result

Python is used to construct our simulation program, and matplotlib was utilized to draw the diagrams representing the evacuation situation. Every second, the current evacuation situation is updated in the last second and displayed via matplotlib. Different kinds of visitors are displayed in different kinds of colors, thus they can be easily recognized in the plot.

Through simulation, it is found out that the total evacuation time of the people on the 2nd floor is 91 seconds with 133 people. The bottlenecks in the 2nd floor are also identified, which is specifically located at the exhibition hall 800, 802, 803 and the door between exhibition hall 802 and 803. At the same time, in Figure 7, the evacuation situation of the 2nd floor at some specific time points during the evacuation process is shown.

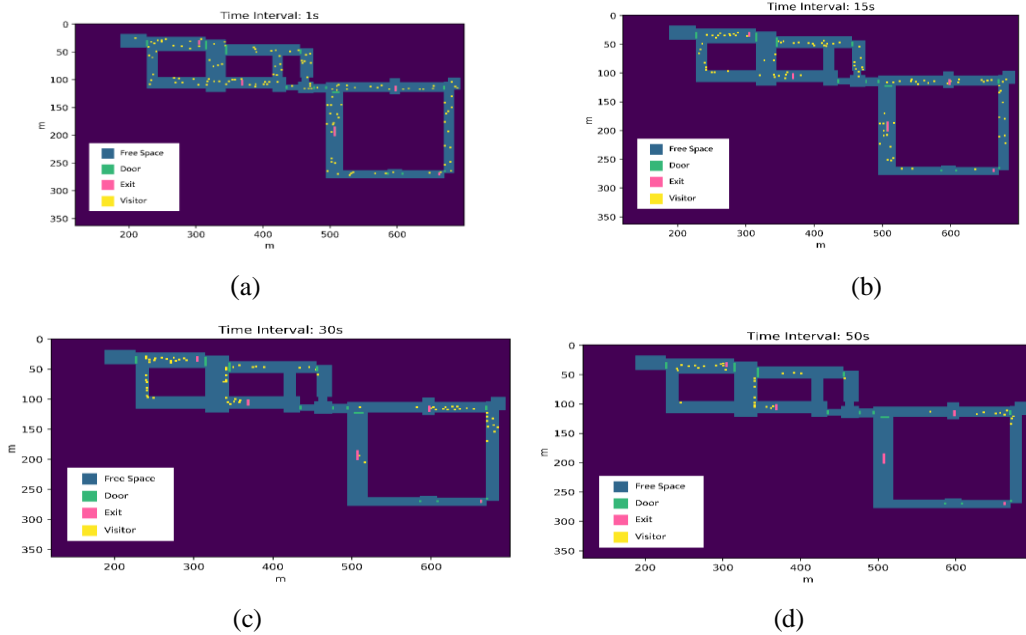


Figure 7: Evacuation situation of the Louvre at

(a) 1st second (b) 15th second (c) 30th second (d) 50th second

With the evacuation process precisely depicted in Figure 7, the congestion along the evacuation route is easily distinguished with naked eye as shown in Figure 8. They are specifically located at the exhibition hall 800, 802, 803 and the door between exhibition hall 802 and 803.

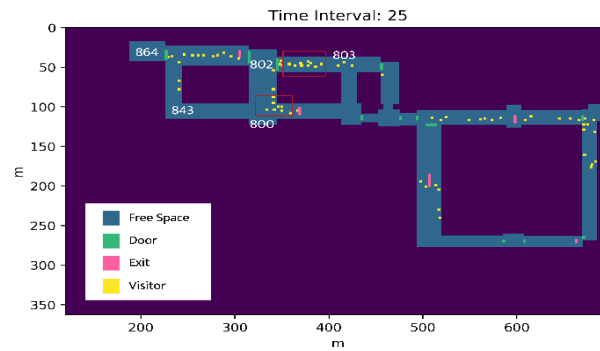


Figure 8: Congestion at 25th second

4. CONCLUSIONS

Based on CA, the model is refined by dividing the area into more dense grids. Combined it with AAC and AAPF which give the individuals the ability to find their way to the doors and let them be alerted by their predecessors who have already get to the exits, a hybrid model is established. This hybrid model is applied to simulate the evacuation process inside the large buildings with complex structure with great intuition and preciseness.

The hybrid model is applied in the Louvre to simulate the evacuation process with real-time data. The evacuation situation during the whole process is obtained, and the bottlenecks of the evacuation route are identified. During the process, the speed difference of the elderly and those at a younger age is taken into consideration. This has proved the high applicability and flexibility of the hybrid model.

The future work could fruitfully explore this issue further by looking into optimizing the speed of AAPF model, and looking into applying the hybrid model to a multi-story building. Also, some research can be done to analyze the influence of stairs and lifts to the hybrid model, so that the hybrid model can be applied to more complex situations.

REFERENCES

- [1] J. SERNA, P. ST. JOHN and R. LIN II, "Disaster after disaster, California keeps falling short on evacuating people from harm's way", 2019. [Online]. Available: <https://www.latimes.com/local/lanow/la-me-paradise-fire-evacuation-system-20181120-story.html>. [Accessed: 24- May- 2019].
- [2] J. Wang, "Uncertainty study on large-scale crowd evacuation in unconventional emergencies", Doctor, University of Science and Technology of China, 2013. (in Chinese)
- [3] X. Zhou, "Analysis of safe evacuation of metro stations", Encyclopaedia Form, vol. 17, p. 263, 2018. [Accessed 24 May 2019]. (in Chinese)
- [4] T. Ping, "Research on Personnel Evacuation Simulation Based on Cellular Automata Model", Computer Simulation, vol. 10, no. 26, pp. 319-322, 2009. [Accessed 24 May 2019].(in Chinese)
- [5] H. Feng, J. Yang, B. Liu and J. Zhu, "Simulation of pedestrian flow evacuation based on cellular automata", Construction Science and Technology, no. 7, pp. 64-66, 2017. [Accessed 24 May 2019]. (in Chinese)

- [6] Q. Du, R. Chen and A. Xu, "Subway pedestrian evacuation model based on ant colony cellular automaton", *Computer era*, no. 2, pp. 18-21, 2018. [Accessed 24 May 2019]. (in Chinese)
- [7] J. Fu, Y. Liu and J. Li, "Fire Dynamic Evacuation Based on Ant Colony Algorithm", *Journal of East China Jiaotong University*, vol. 34, no. 3, pp. 118-124, 2017. [Accessed 24 May 2019]. (in Chinese)
- [8] N. Ta, "Modelling and simulation of pedestrian evacuation based on spatial refinement cellular automata", Master, Inner Mongolia university, 2016. (in Chinese)
- [9] R. He, "Simulation of crowd evacuation based on potential energy field model", Master, SunYat-sen University, 2019. (in Chinese)
- [10] Z. JIN, X. Ruan and L. Li, "Evacuation Simulation in Narrow Passage Under Fire Scenario Based on Cellular Automaton", *Journal of Tongji University(Natural Science)*, vol. 46, no. 8, pp. 1026-1034, 2018. [Accessed 24 May 2019]. (in Chinese)
- [11] C. Langton, "Self-reproduction in cellular automata", *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 135-144, 1984. Available: 10.1016/0167-2789(84)90256-2.
- [12] D. Li, L. Yuan, Y. Hu and X. Zhang, "Large-scale crowd motion simulation based on potential energy field", *Journal of Huazhong University of Science and Technology(Nature Science Edition)*, vol. 44, no. 6, pp. 117-122, 2016. [Accessed 24 May 2019]. (in Chinese)
- [13] Z. Wu, "Evacuation simulation of crowds in unfamiliar environments", Master, SunYat-sen University, 2010. (in Chinese)
- [14] Y. Yan, "Research and application of ant colony algorithm", *China Venture Capital*, no. 3, pp. 201-202, 2019. [Accessed 24 May 2019]. (in Chinese)
- [15] Q. Wu and L. Wang, *Intelligent Ant Colony Algorithm and Application*. Shanghai: Shanghai Science and Technology Education Press, 2004. (in Chinese)
- [16] "Affluences - L'affluence en temps réel", *Affluences*, 2019. [Online]. Available: <https://www.affluences.com/louvre.php>. [Accessed: 24- May- 2019].
- [17] T. Reporters, "Terror attacks in France: From Toulouse to the Louvre", *The Telegraph*, 2018. [Online]. Available: <https://www.telegraph.co.uk/news/0/terror-attacks-france-toulouse-louvre/>. [Accessed: 24- May- 2019].
- [18] "Online Extra: Q&A with the Louvre's Henri Loyrette", *Bloomberg.com*, 2002. [Online]. Available: <https://www.bloomberg.com/news/articles/2002-06-16/online-extra-q-and-a-with-the-louvres-henri-loyrette>. [Accessed: 24- May- 2019].
- [19] "The "Pyramid" Project (2014-2016)", *Louvre.fr*, 2014. [Online]. Available: http://www.louvre.fr/sites/default/files/dp_pyramide%2028102014_en.pdf. [Accessed: 24- May- 2019].
- [20] "Louvre - Interactive Floor Plans | Louvre Museum | Paris", *Louvre.fr*, 2019. [Online]. Available: <https://www.louvre.fr/en/plan>. [Accessed: 24- May- 2019].

AUTHORS**Corresponding author:**

Xiongda Chen, professor, majored in operational research and cybernetics.

First Author Equally:

Hao Yuan, undergraduate, majored in the study and research of digital media.

Guo Yu, undergraduate, majored in the study and research of mathematical statistics.

Yifan Ma, undergraduate, majored in the study and research of software management.

Second Author:

Jieneng Chen, undergraduate, majored in the study and research of computer science

COLLABORATIVE AND FAST DECRYPTION USING FOG COMPUTING AND A HIDDEN ACCESS POLICY

Ahmed Saidi¹, Omar Nouali² and Abdelouahab Amira³

¹²³Department of Computer Security, Research Center for Scientific and
Technical Information, Algiers, Algeria

¹³Faculty of Exact Sciences, Universite de Bejaia, 06000 Bejaia,
Algeria.

ABSTRACT

Nowadays, IOT (Internet Of Things) devices are everywhere and are used in many domains including e-health, smart-cities, vehicular networks,.. etc. Users use IOT devices like smartphones to access and share data anytime and from anywhere. However, the usage of such devices also introduces many security issues, including in data sharing. For this reason, security mechanisms such as ABE (Attribute-Based Encryption) have been introduced in IOT environments to secure data sharing. Nevertheless, Ciphertext-Policy ABE (CP-ABE) is rather resource intensive both in the encryption and the decryption processes. This makes it unadapted for IOT environments where the devices have limited computing resources and low energy. In addition, in CP-ABE, the privacy of the access policy is not assured because it is sent in clear text along with the cipher-text. To overcome these issues, we propose a new approach based on CP-ABE which uses fog devices. The letters collaborate to reduce the bandwidth, and partially delegates data decryption to these fog devices. It also ensures the privacy of the access policy by adding false attributes to the access policy. We also discuss the security properties and the complexity of our approach. We show that our approach ensures the confidentiality of the data and the privacy of the access policy. The complexity is also improved when compared with existing approaches.

KEYWORDS

Fog Computing, Access Control, Attribute based Encryption, Decryption Outsourcing.

1. INTRODUCTION

Fog computing is an emerging paradigm that extends cloud computing. It acts as an intermediary between the cloud and end devices by bringing processing, storage and networking services closer to these end devices[1]. For example, the processing and the storage of temporary data which are collected by sensors can be delegated to the hospital local servers which act as fog nodes. This architecture allows to reduce the amount of data transferred to the cloud for processing, analysis and storage. As a consequence, the network traffic bandwidth and latency are reduced. This is especially the case of data sharing, which allows users to store their data, access it from anywhere, at anytime, and share it with other users. However, Users lose control over their data when it is outsourced to the Cloud or when it is processed by fog nodes.

To address these issues, it is essential to use mechanisms such as encryption and decryption, which allow to secure data sharing.

ABE [2] is a new, efficient and promising encryption/decryption technique that aims to achieve scalable and fine-grained access control. It keeps the encrypted data confidential even when the storage server is untrusted. In ABE, the encryption is based on a set of attributes describing data properties, user properties and properties of the environment, as well as an access structure indicating who can access what. ABE is constructed from an access tree representing a logical expression that combines several attributes via AND and OR operators. There are two main variants of ABE: (1) ABE Key-Policy (KP-ABE) [3] and (2) ABE-Ciphertext-Policy (CPABE) [2]. The KP-ABE, the encrypted data is associated with a set of attributes. Whereas, the key is associated with the access policy. The users can decrypt the data if and only if the attributes in the data satisfy the access policy. On the other hand, in CP-ABE, the attributes are associated with the user's private key and the data is encrypted with the access policy.

Nevertheless, one of the drawbacks of ABE is that the computational cost during in the encryption and decryption phases increases exponentially with the complexity of the access policy. This is a considerable limitation when devices are limited in terms of resources (for example CPU, energy, etc.). Another drawback of ABE is that the access policy is sent in clear text along with the ciphertext. A malicious user can obtain both the ciphertext and the associated access policy. The latter contains some sensitive information (like social security number, name, etc), that can be exploited to compromise the legitimate user's privacy.

In this paper, we propose a new solution based on CP-ABE. Our approach uses fog nodes collaboration and a newly proposed partial decryption approach with a hidden access policy to achieve low computation overhead and achieve secure and fine access control.

The basic idea of our approach is as follow:

(1) We use Fog nodes to offer for fast and more convenient computing services. Moreover, the fog computing provides low-latency communications.

(2) Our scheme delegates the user's attributes and the decryption operation to the fog nodes without revealing the original message, the set of user's attributes or the attributes in the access policies to the fogs. Fog nodes collaborate with each other to help the user decrypt the data. To delegate the decryption operation, the TA (Trusted Authority) creates intermediate keys for the fog nodes using the user's secret key. This intermediate key is used by the fog to partially decrypt the text without revealing which attributes are used in the decryption process.

(3) We add false attributes to hide the access policy. The trusted authority divides the set of attributes over all available fogs. Each fog manages its own set of attributes. When user (Data Owner) creates an access policy, he divides the access policy and adds false attributes to each subtree of the access policy. This operation is performed by taking into account the number of available fog and according to the set of attributes managed by Fog node. In this way, the fog nodes will not be able to deduce which attributes participated in the decryption phase.

Contribution:

The main contributions of this paper are as follow:

- To the best of our knowledge, our work is the first to hide and protect user attributes against fog nodes in outsourced decryption process phase using fog nodes.

- We present a secure outsourcing and a fast decryption approach by delegating heavy computations from IOT to Fog. This is performed by creating an intermediates key from the user which are used to partially decrypt the ciphertext. This means that the computational decryption complexity of IOT is independent of the number of attributes.
- We divide the set of universal attributes by the set of available fogs so that each fog manages its proper attributes. When the data owner wants to encrypt the data, the access policy is divided according to the attributes that each fog manages.
- We extend our approach by adding false attributes for to each access policy so that fogs nodes cannot deduce the real attributes. In addition, fog nodes are not able to deduce the valid attributes users in the decryption process even if the case where fog nodes are compromised or collude.
- We thoroughly analyze the security properties and the decryption complexity of our proposed scheme.

Paper organization:

The reset of this paper is organized as follows. In Section II, we examine the existing solutions that aim to reduce the encryption and decryption costs. In Section III, we introduce the system and threat models. In Section IV, we give a high-level overview of the proposed scheme. The detailed construction of our ABE based outsourced decryption scheme is given in Section VI. We analyze the security and complexity of our approach in Section V. In section VII, we introduce some typical scenarios where our proposed scheme can be applied. The paper is concluded in Section VIII.

2. RELATED WORKS

Attribute-Based Encryption (CP-ABE) [2] is considered one of the most appropriate technologies for performing fine-grained access control. However, the encryption and decryption processes in this scheme are very complex and time consuming. In order to reduce the cost of encryption and decryption at the user level, several schemes for externalizing the computation were proposed.

Zhou et al. [4] proposed a new CP-ABE scheme, in which the encryption and decryption process is outsourced on external cloud based services. In the encryption process, the authors connect two access structures T1 and T2 to form a single access policy. A root AND node connect these access policies. The first part of the encrypted text is generated by sending T1 to an external encryption service while the second part is computed by the user using T2, where this T2 contains only one attribute. However, one flaw in this approach is that the access policy in this scheme is not hidden.

In their work Touati et al. [5] present a cooperative CP-ABE for the Internet of Things, where the complex operations of the CPABE encryption primitive forced authors to use intermediates Unconstrained devices to outsourced encryption process. The authors assume that unconstrained devices are trusted. In this scheme, the data owner (device A that is a resource constrained device) encrypts the data under access T. During the process; device A is supported by a set of secure assistant devices that perform the exponentiation operation instead of the device A itself. The authors suppose that the intermediate unconstrained devices are trusted, but they do not suggest externalization of the decryption process, Another drawback is that the

access policy is sent in the clear on the network, where the access structure can also contain some sensitive private information.

In [6], the authors propose a new method for outsourcing CP-ABE, namely the EOEB (outsourcing mechanism for the encryption of the ABE encryption policy). The main idea is to reduce encryption costs by delegating the most intensive computations of the encryption phase of the CP-ABE to a semi-trusted party. The authors divide the encryption process in to two phases: the Pre-delegation phase, and the compDelegation phase. Pre-delegation is performed by KDG (Key delegation) which executes the configuration algorithm as in the basic CP-ABE. It also generates a secret delegation key for each data producer (DP) and a list of security parameters. This list is then sent to DG (delegate). Two steps are executed in the compDelegation phase. The first step is executed by DP. In this step, the DP generates the temporal encrypted text CT' which contains the Blinded value s . The second step is executed by DG (delegate) which executes the most expensive computation operation without any knowledge of the secret message M . Nevertheless as in the work of [5], the authors do not propose to outsource the decryption process which consumes IOT energy at the user level and they do not hide the access policy.

Fan et al. [7] proposed an outsourced, secure and verifiable multi-authority access control system called VO-MAACS. In their construction, most encryption and decryption computations are outsourced to Fog devices, and the result can be verified by signed the message. The Fog devices are responsible for the transmission of data. They are also responsible for a part of the computation of encryption and decryption. Fog devices can help data owners to generate some of the encrypted text. They also help DVs to decrypt some of the encrypted text but only when the DV attributes satisfy the access policy. In this proposal, the authors used a secret linear sharing scheme (LSSS) to construct an access policy. Despite this, in their scheme, the access policy is not hidden.

In [8] propose a CP-ABE scheme with a hidden access strategy and fast decryption that improves the decryption efficiency at the user level. The authors also propose a method to hide the access policy by adding false attributes to the access policy which preserves its confidentiality. This method ensures fast decryption and hidden access policy. However, in this scheme even if there is an improvement in energy consumption in the IOTs. The decryption process still is the energy intensive since it is executed at the user level.

In [9] the authors proposed Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT (PHOABE), in this scheme, the attributes in access policy are hidden, and the decryption process is outsourced to the third party. However, the solution is proven selectively secure and even though the decryption process is outsourced the overhead cost at user still important Thus we rely on the work of Wang and Lang [8] where we have modified their scheme for outsourcing of the decryption process to several Clouds, using intermediate keys for partial decryption of the data.

In existing approaches, the policy is not hidden in other works energy intensive and selective privacy methods is used.in other cases, outsourcing is not assured.

3. BACKGROUND

In this section, we present some notation used in this article. Then we illustrate the details of the encryption and decryption process.

A) Preliminaries

- 1) **Composite-Order Bilinear Group:** Let G denote an algorithm that takes as input a security parameter and outputs a tuple $(N = p_1 p_2 p_3 p_4, G, G_T, e)$, where $p_1 p_2 p_3 p_4$ are distinct primes, G and G_T are cyclic groups of order N , And $e: G \times G \rightarrow G_T$ is a bilinear map such that:

- (Bilinear) $\forall g, h \in G$ and $x, y \in Z_N$, it satisfies $e(g^x, h^y) = e(g, h)^{xy}$.
- (Non-degenerate) $\exists g \in G$ such that $e(g, g)$ has order $N \in G_T$.

We require that the group operations in G and G_T and the bilinear map e are all computable in polynomial time. Let Gp_1, Gp_2, Gp_3 and Gp_4 denote the subgroups of G with orders p_1, p_2, p_3 and p_4 , respectively. Note that if $g_i \in Gp_i$ and $g_j \in Gp_j$ for $i = j$, then $e(g_i, g_j) = 1$. If the generator of Gp_j is $g_i (i \in \{1, 2, 3, 4\})$, then every element $h \in G$ can be expressed as $g_1^{a_1} g_2^{a_2} g_3^{a_3} g_4^{a_4}$ for some values $a_1, a_2, a_3, a_4 \in Z_N$.

- 2) **Access Tree:** Let T be a tree representing an access structure. Each non-leaf node of the tree represents a threshold operator, which is described by its children and a threshold value. If num_x is the number of children of node x , and k_x is its threshold value, then $1 \leq k_x \leq num_x$. When $k_x = 1$, the threshold is an OR operator, and when $k_x = num_x$, it is an AND operator. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$ [2]. Let T be an access tree with root r . The subtree of T rooted at node x is denoted by T_x . Thus, T is the same as T_r . If a set of attributes ω satisfies the access tree T_x , we denote it as $T_x(\omega) = 1$. We compute $T_x(\omega)$ recursively as follows: If x is a non-leaf node, we evaluate $T_x(\omega)$ for each child x of node x . $T_x(\omega)$ returns 1 if and only if at least k_x children return 1. If x is a leaf node, then $T_x(\omega)$ returns 1 if and only if $att(x) \in \omega$, where $att(x)$ denotes the attribute associated with node x [2].

B) CP-ABE Algorithms

CP-ABE consists of the following algorithms [2]:

- **Setup (U).** This algorithm takes as input an attribute universe U . It will initialize the system and generate the master key MK and the public key PK .
- **KeyGen (PK, MK, ω).** This algorithm takes as input the public key PK , the master key MK and a users attribute set ω . It will output a private key SK_ω .
- **Encryption (PK, M, T).** This algorithm takes as input the public key PK , a message M and an access-policy tree T . It will produce a ciphertext CT .
- **Decryption (SK_ω, CT).** The decryption algorithm takes as input a private key SK_ω and a Ciphertext CT . It will output the plaintext M if ω satisfies T .

4. SYSTEM MODEL AND THREAT MODEL

A. SYSTEM MODEL

We consider a file sharing system consisting of five parties: Trusted Authority (TA), Data Owner (DO), Data User, Fogs and the Cloud. In this system, the TA is responsible for

system initialization, authenticating the users' attributes, creating and sending the secret keys to the users and the generating intermediaries keys to the fogs. The Data owner is the user who wants to upload and share his data; it is also his role to specify the access policy which is used to encrypt the data. The policy is used to control who can access to this shared data. The data user is the one who wants to access to the shared data; he solicits the TA by sending his attributes in order to obtain a private key that will be used to decrypt the data. The cloud provides a storage service to users so that the shared data can be accessed anywhere and anytime. Fogs are entities to that collaborate and help users to partially decrypt the data. The system model is shown in Figure.1.

Our proposed model secure data sharing system by using following functions.

1. $\text{Setup}(\tau, N, f) \rightarrow \{PK, MSK\}$: the TA executes the setup function which takes as input a security parameter τ , the set of universal attributes N and the number of available Fogs f . As a result, it outputs a public key (PK) and a master secret key (MSK). The public key PK is known by all entities of the system.
2. $\text{Keygen}(PK, MSK, S, F) \rightarrow \{SK, TK\}$: this function is executed by the TA. When the user requests his private key by sending his set of attributes S , the TA generates two keys, a secret key (SK) and an intermediate key (TK). The latter operation is used after checking the validity of the attributes. The SK key is sent to the user while the TK key is sent to the fogs (F) and is used in the partial decryption phase. Sending these keys is performed through secure channels.
3. $\text{Encryption}(PK, M, T, L) \rightarrow \{E, CT_i\}$: the user encrypts the message M by specifying an access policy in tree form (T) and outputs a ciphertext (E), in addition to several sub-tree CT_i according to the available fogs list L .
4. $\text{DecrypPartial}(CT_i, TK_i) \rightarrow \{C_i, P_i\}$: using its intermediate key TK , the fog partially decrypts the ciphertext $\{C_i, P_i\}$ are sent to the user.
5. $\text{Decryption}(C_i, P_i, SK, E) \rightarrow M$: the user decrypts the ciphertext CT using $\{C_i, P_i\}$ which are sent by all the fogs. By using his private key, the user can recover the message M .

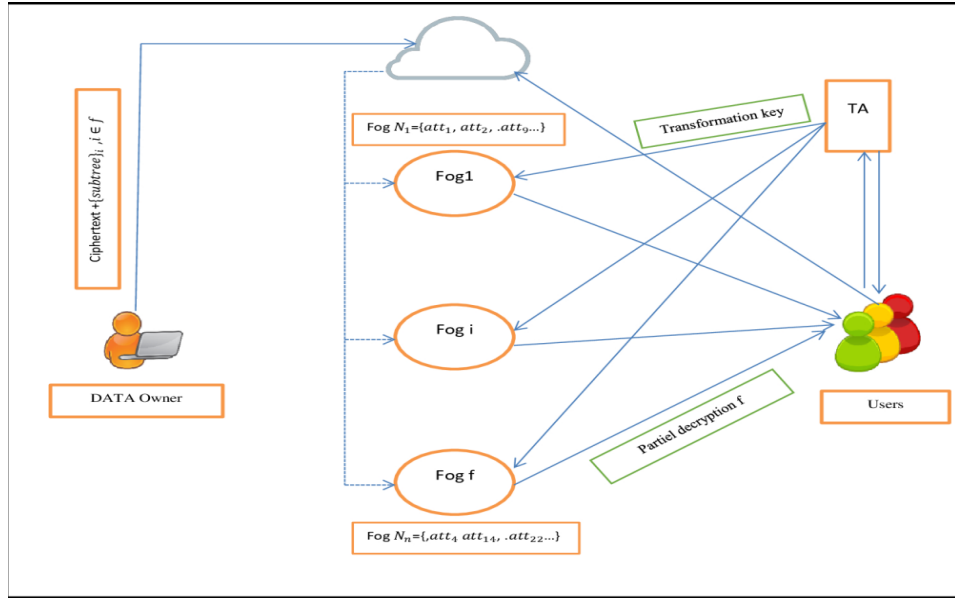


Figure 1. scheme of the proposed solution

B. THREAT MODEL

In our proposed system, we assume that TA is a trusted entity as in any system that uses ABE. We also assume that cloud and fogs are semi-trusted entities ie: the cloud and the fogs apply the protocols but curious entities. Also, we suppose that each fog manages a set of attributes in such a way that: $\forall i \neq j N_i \cap N_j = \emptyset$ where N_i is the set of attributes belonging to the fog_i . The communication between the entities is secure.

5. DESCRIPTION OF THE PROPOSED APPROACH

In this section, we give a description of the different phases of our approach.

- (A) Initialization phase : in this phase, the trusted authority generates two keys, a public key (PK) that is shared for all entities in the system and a master key (MSK) that will be kept secretly. After creating the keys, the TA assigns each user its own attributes. At the end of this step, each user will know the public key and the sets of all the attributes in the system.
- (B) Encryption phase: when the Data Owner (DO) wants to share information with another user in the system according to an access policy, he creates an access policy T in tree form. He divides this tree into several subtrees T_i according to the available fogs and by taking into account the attributes managed by each fog. The list of available fogs and their attributes is sent by the TA (Figure. 2). After obtaining the subtree T_i , the DO adds false attributes to hide the real attributes. He chooses random numbers $\{s_1 \dots s_f\}$ corresponding to each fog $\{Fog_1 \dots Fog_f\}$, where s_i is shared by all the attributes in T_i . Each s_i is shared for each node of the access tree T_i . The secret s_i is divided according to the "Top-Down approach" where the secret s_i is divided by $(t - n)$ Shamir secret sharing approach from the root to the leaf node. Where the parameter n is number of all child nodes and t is number of child nodes for recover secret s_i . Each real attribute in T_i will contain the shared λ_i of s_i . In contrast, the false attributes will not contain the share λ_i of s_i , moreover,

the false attributes will be eliminated in the partial decryption phase. After that, the DO sends the ciphertext with all T_i to the cloud for storage.

- (C) Decryption phase: this phase contains two phases: the partial decryption and the final decryption. In the partial decryption, When a user wants to access to the shared data, he requests his private key from TA with his attributes (S). The TA chooses two random variable θ and t , where ($SK = \theta$) will be the private key of the user and t it used with the set of users attributes to create the transformation keys TK_i for each available fog_i . The fogs can use TK to decrypt the data partially. Both keys are sent securely. The partial decryption at the level of fog_i is performed with the TK_i key. Each fog_i decrypts the data partially without knowing which attribute participated in the partial decryption. Each fog sends partially decrypted data to the user to recover the message M . Finally in the final phase and After the user receives all partially decrypted data, he recovers the message with his private key SK .

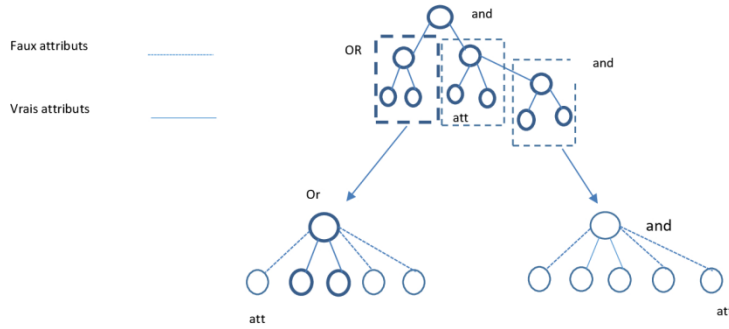


Figure 2. Division of access policy into several subtrees.

6. CONSTRUCTION

The algorithm starts by the setup phase until decryption phase. There are six phases .Each phase is detailed in the following paragraphs:

Initialization phase

- (A) **Setup** (τ, N, f) : the algorithm takes as input a security parameter τ , the set of universal attributes N and the number of available Fogs f . The algorithm chooses a bilinear group G with an order $O = p_1 p_2 p_3 p_4$, for each attribute $A_i \in N (1 \leq i \leq n)$ where n is the number of attributes in the universe N . Then is selects $h_i \in Z_N^*$ and finally selects a random element $\alpha, \beta \in Z_N^*$ and $g \in G_{p_1}$. The public key is defined by:

$$PK = \{N, g, y = (g, g)^\alpha, L = g^\beta, H_i = g^{h_i} (1 \leq i \leq n)\}$$

and the master key by:

$$MK = \{\alpha, \beta\}.$$

- (B) The algorithm divides the set N by the number of available fogs f . This means $N = N_1 \cup N_2 \cup \dots \cup N_f$ in such a way $\forall i \neq j, N_i \cap N_j = \emptyset$ where N_i is the set of attributes belonging to the fog_i This phase is executed by the trusted authority (noted TA in **Figure. 1**).

- (C) When the user requests his private key with his set of attributes. The TA chooses a random variable θ that will be the private key of the user($SK = \theta$).

Encryption phase

(A) In this phase, the *DO* executes the Encryption primitive denoted **Encryption**(PK, M, T, L) as follows: The Encryption algorithm takes as input the public key PK , the message M and the access policy T in the tree form and L which represents the list of available Fogs with their attributes(N_i).

(B) The algorithm divides the tree T into several subtrees T_i according to the number of available Fogs. Each subtree will contain the attributes of the destination Fog.

(C) The Sender adds false attributes(nodes) to the subtrees according to the universe of attributes of the destination Fog. Let U_i be the set of attributes of the subtree T_i after adding false attributes. In the subsequent step, The Sender chooses a random numbers $\{s_1 \dots s_f\}$ corresponding to each fog $\{Fog_1 \dots Fog_f\}$ where s_i is shared by all the attributes in T_i .

(D) The algorithm shares the secret s_i as follows: a polynomial $q_i(x)$ degree $k_i - 1$ is chosen for each node (including the leaf node) in T_i where $k_i = |T_i|$ (number of elements in(T_i)). These polynomials are generated in a recursive manner starting from the root node r . We define $q_{ir}(0) = s_i$ (where r represent the root node in the tree) then other value $k_i - 1$ are defined randomly to complete the construction. Once all the polynomials have been defined we put $\lambda_{xi} = q_{xi}(0)$ for each node x in T_i , we choose random elements $Z_0, \{Z_i\}_{A_i \in U_i} \in G_{p^4}$ knowing that $att(x) = A_i$ and $index(y)$ is attributes index of y in T_i , the ciphertext is generated as follows:

$$CT = \{E = My^s, E_0 = g^s Z_0, CT_i\}$$

$$CT_i = \left\{ \begin{array}{l} \forall A_i \in T_i: E_i = L^{\lambda_{xi}} H_i^{s_i} Z_i \\ \forall A_i \notin T_i: E_i = H_i^{s_i} Z_i \end{array} \right\}, T_i$$

Where $s = \sum s_i$. The ciphertext is formed as CT include CT_i that is stored in the cloud.

Decryption phase

(A) When a user wants to access the shared data, he sends a request to the cloud about the encrypted data and request the *TA* to create the transformation keys TK . So, the *TA* executes the primitive **KeyGen**(PK, MK, S, θ, f) as follows: the *TA* (Trusted authority) creates the transformation keys TK for each fog. For that, the *TA* starts the key generation procedure where this key makes it possible to perform a partial decryption. To create TK **KeyGen** chooses a random element $t \in Z_N^*$ and $R, R_0, \{R_i\}_{A_i \in S_i} \in G_{p^3}$, then returns the transformation key for each Fog. Formally:

$$TK = \{D = g^{(\alpha - \beta t)\theta} R, D_0 = g^t R_0, \forall A_i \in S_i: D_i = H_i^t R_i\}$$

Finally, the *TA* distributes the TK_i key to fog_i .

(B) Upon receipt of the TK_i key and CT_i , fog_i executes the following function:

DecrypPartila(CT_i, TK_i) : This algorithm takes as input CT_i and TK_i . When the fog_i receives CT_i it uses its transformation key TK_i to partially decrypt the ciphertext. Two recursive functions are used:

DecryptNode_CT_i(CT_i, x) which takes as input CT_i and the node x which belongs to T_i.
DecryptNode_TK_i(TK_i, x) which takes as input the transformation key and the x node.

The algorithm of **DecryptNode_CT_i** and **DecryptNode_TK_i** is defined by the following instructions:

If the node x is a leaf node Set *DecryptNode_CT_i(CT_i, x) =*

$$E_i = \begin{cases} L^{\lambda_{xi}} H_i^{s_i} Z_i & \text{if } A_i \in T_i \\ H_i^{s_i} Z_i & \text{if } A_i \notin T_i \end{cases}$$

$$\text{DecryptNode_TK}_i(\text{TK}_i, x) = D_i = H_i^t R_i$$

We consider the case where x is an internal node. The two functions *DecryptNode_CT_i* , and *DecryptNode_TK_i* are executed in the following way: (knowing that the direction of execution is root to down) For each node y that is the child of x *DecryptNode_CT_i* and *DecryptNode_TK_i* are invoked. The result is saved respectively in F_y and K_y, let Q_x a set of y nodes child that belongs to T_i and Q_{x'}, the set of y nodes that does not belong to T_i. We have Q_x ∪ Q_{x'} = all the children of the x in the T_i tree. If y is a node then we calculate:

$$\begin{aligned} F_x &= \prod_{y \in Q_x \cup Q_{x'}} F^{l_y v_x(0)} \\ &= \prod_{y \in Q_x} (L^{\lambda_{xi}} H_i^{s_i} Z_i)^{l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} (H_i^{s_i} Z_i)^{l_y v_x(0)} \\ &= \prod_{y \in Q_x} g^{\beta \lambda_{yi} \cdot l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} H_i^{s_i \cdot l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} Z_i^{l_y v_x(0)} \\ &= g^{\beta \lambda_{xi}} \cdot F_{x,1} \cdot F_{x,2} \end{aligned}$$

And

$$\begin{aligned} K_x &= \prod_{y \in Q_x \cup Q_{x'}} K^{l_y v_x(0)} \\ &= \prod_{y \in Q_x \cup Q_{x'}} H_i^{t \cdot l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} R_i^{l_y v_x(0)} \\ &= K_{x,1} \cdot K_{x,2} \end{aligned}$$

If the node is non-leaf node we calculate:

$$\begin{aligned} F_x &= \prod_{y \in (Q_x \cup Q_{x'})} (g^{\beta \lambda_{yi}} \cdot F_{x,1} \cdot F_{x,2})^{l_y v_x(0)} \\ &= \prod_{y \in Q_x} (g)^{\beta \lambda_{yi} \cdot l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} F_{y,1}^{l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} F_{y,2}^{l_y v_x(0)} \\ &= g^{k \lambda_{xi}} \cdot F_{x,1} \cdot F_{x,2} \end{aligned}$$

And

$$\begin{aligned} K_x &= \prod_{y \in Q_x \cup Q_{x'}} K^{l_y v_x(0)} \\ &= \prod_{y \in Q_x \cup Q_{x'}} K_{y,1}^{l_y v_x(0)} \cdot \prod_{y \in Q_x \cup Q_{x'}} K_{y,2}^{l_y v_x(0)} \\ &= K_{x,1} \cdot K_{x,2} \end{aligned}$$

In the previous equation, we have F_{x,1} = K_{x,1} the parameter v_x = {index(y)/y ∈ (Q_x ∪ Q_{x'})} and l_yv_x(0) is the coefficient of lagrange. If we call both functions from root r of T_i then we obtain:

$$A = \text{DecryptNode_CT}_i(\text{CT}_i, r) = g^{ks_i} \cdot F_{r,1} \cdot F_{r,2}$$

And

$$B = \text{DecryptNode_TK}_i(\text{TK}_i, r) = K_{r,1} \cdot K_{r,2}$$

We calculate:

$$\begin{aligned} C_i &= e(A, D_0) / e(E_0, B) \\ &= e(g^{\beta s_i} \cdot F_{r,1} \cdot F_{r,2}, g^t R_0) / e(g^{s_i} Z_0, K_{r,1} \cdot K_{r,2}) \\ &= e(g^{\beta s_i}, g^t) \cdot e(F_{r,1}, g^t) \cdot e(F_{r,2}, g^t) \cdot \\ &\quad e(g^{\beta s_i} \cdot F_{r,1} \cdot F_{r,2}, R_0) / e(g^{s_i}, K_{r,1}) \cdot e(g^{s_i}, K_{r,2}) \cdot e(Z_0, K_{r,1} \cdot K_{r,2}) \\ C_i &= e(g, g)^{\beta t s_i}. \end{aligned}$$

And also:

$$\begin{aligned} P_i &= e(E_0, D) \\ &= e(g^{s_i} Z_0, g^{(\alpha - \beta t)\theta} R) \\ &= e(g^{s_i}, g^{(\alpha - \beta t)\theta} R) \\ &= e(g, g)^{s_i(\alpha - \beta t)\theta} \end{aligned}$$

(C) Finally, the fog sends the partial decryption C_i and P_i to the user.

(D) Upon receipt of all shares parts of fog_i , the user executes the following function:

Decryption(C_i, P_i, SK, E) : this algorithm is executed by the user. If the user receives all the parts which are partially decrypted from the Fog, then he knows that his attributes satisfy the access policy

Otherwise, he rejects the decryption. When the user receives all the parts which was partially decrypted, he uses his private key $Sk = \theta$ and the ciphertext transformed by the Fog (C_i, P_i) to recover the original message.

Formally:

$$\begin{aligned} \frac{E}{(\prod P_i)^{\frac{1}{\theta}} \cdot \prod C_i} &= \frac{E}{(\prod e(g, g)^{s_i(\alpha - \beta t)\theta})^{\frac{1}{\theta}} \cdot \prod e(g, g)^{\beta t s_i}} \\ &= \frac{E}{(e(g, g)^{(\alpha - \beta t)\theta \sum s_i})^{\frac{1}{\theta}} \cdot e(g, g)^{\beta t \sum s_i}} \\ \frac{E}{(e(g, g)^{s(\alpha - \beta t)\theta})^{\frac{1}{\theta}} \cdot e(g, g)^{s\beta}} &= \frac{E}{e(g, g)^{s(\alpha - \beta t)} \cdot e(g, g)^{s\beta}} = \frac{E}{e(g, g)^{s\alpha}} = \frac{M e(g, g)^{s\alpha}}{e(g, g)^{s\alpha}} = M \end{aligned}$$

7. ANALYSES

In this section, we discuss the security properties of the proposed solution involving data privacy, fine-grained access control, and collision resistance.

(A) Security Proprieties

(1) **Data confidentiality**: The confidentiality requires that the cloud and the fog cannot learn the encrypted data, in the decryption-outsourcing algorithm; the cloud is responsible only for storing the encrypted data as $M \cdot e(g, g)^s$ where s is kept secret by the user. While, the Fogs are

responsible only for the partial decryption of the data, and since the transformation keys TK_i are generated by TA with the secret key of the user, only the end user where his attributes correspond to the access policy, can recover the encrypted data, in other words, fogs cannot recover random value s where this value is divided among the fogs in the encryption process even if the fogs cooperate with each other, since in the processing of the partial decryption the S_i are blinded with the secret key of the user $SK = \theta$. Thus, we conclude that our scheme is secure in protecting the confidentiality of the message.

(2) **Hidden access policy:** in our schema the DO adds false attributes to the access policy, with this method the malicious users and even the Fogs cannot have the real attributes even if the Fogs cooperate with each other, this will lead to the addition of several false attributes which further complicates the task of having the right attributes. Also, the Fogs and even the users cannot know which attribute participated in the decryption of the data as all the attributes of the users whether they belong to the access policy are being applied in the decryption process.

(3) **Fine-grained access control:** the proposed solution uses the CP-ABE algorithm, where the DO defines an access policy for each outsourced data. This access policy is in the form of an And-gate tree where the tree contains the attributes that allow access to the data, in this way only the users that their attributes match with the attributes in the access policy can decrypt the data.

(4) **Collusion resistance:** the collision resistance is the property that the CP-ABE assumes. In our solution, the algorithm Keygen generates a different random values t for each user and which is integrated into the key of transformation. It means that each key of the user is randomized; this means that users cannot combine their keys to decrypt the data, so malicious users cannot collaborate to expand their access privileges including fog nodes since the transformation key contain the random value t .

(A) Analysis and discussion:

An overview comparison of some existing CP-ABE schemes with our scheme is presented in Table1 .Our scheme achieves policy hiding ,fast decryption, outsourced decryption process and proven fully secure in the standard model . The access policy is specified based on the tree structure which allows the data owner to specify a complex access policy in intuitionistic form, There by delivering a better user experience than LSSS. The comparison indicates that our scheme has all of the following features: hidden policy, fast decryption, outsourced decryption, expressivity and full security.

Table1. an overview comparison

Scheme	Access Structure	Policy hidden	Fast decryption	outsourced computation	security
[4]	Tree	No	No	Yes	Selective
[5]	Tree	No	No	Yes	Selective
[6]	Tree	No	No	Yes	Selective
[7]	LSSS	No	No	Yes	Selective
[8]	Tree	Yes	Yes	No	Full
[9]	LSSS	Yes	No	Yes	Selective
Our approach	Tree	Yes	Yes	Yes	Full

(B) Performance Analysis

In this section, we compare our scheme against two approaches: (1) traditional CP-ABE and (2) the scheme proposed by Wang and Lang [8]. Our comparative study, illustrated in Table 2, is based on the decryption complexity at the user level. This choice is motivated by the fact that we used partial decryption that is delegated to Fogs which have unlimited recourse in terms of energy and computing capacity.

Table2. computation cost

Scheme	Complexity of decryption
CP-ABE	$(2n+1)P+2M$
[8]	$3P+2(n)E$
Our schema	$1 E+2(F)M$

Modular exponentiation (E) and bilinear pairing (P) are two computationally expensive operations in attributes based encryption. We utilize the number of E and P as measurements to evaluate the performance of our scheme. According to Table 1, we see that the decryption cost in the traditional CP-ABE scheme is significant. The user executes $(2n + 1)$ pairing operation (P), where n the number attributes in the access policy. In addition, the user also executes $2M$ Where M denotes the multiplication group in G. Unlike the approach in [8] where the user executes $3P$ and $2(n)E$. However, in our scheme, the user executes only $1E$ exponentiation and $2(F)M$ and the fog execute $3p + 2(n)E$ where n denote the number of attributes in CT_i and F denotes the number of available Fogs in the scheme. Knowing that multiplication consumes less than paring and exponentiation operations we notice that our scheme improved of decryption at the user level compared to other scheme mentioned above. However, the increase of computation on fog side which should be insignificant for the fog.

8. APPLICATION SCENARIOS

In this section, we introduce an application scenario. Our schema can be used in healthcare systems, where wearable devices can detect and collect users health data. The system is composed of entities such as medical insurance, analysis laboratory, private hospitals, hospitals, where each entity manages a set of attributes. In addition, each entity is connected to a fog that will manage these attributes. A doctor or a member of the patient's family is authorized to decrypt the data (according to the access policy). Because the doctor or the family member use constrained devices, they request the fog to partial decrypt the data. According to our proposed method, the scenario is defined as follows:

- 1) After receiving the collected data on his smartphone, the patient (or data owner) defines -by utilizing an application GUI for example- the access policy which specifies who can access the data. for Example a doctor .
- 2) Then the device splits the access policy by taking into consideration attributes that are managed by each fog. Next, it pads them with false attributes and sends each part to the corresponding fog.
- 3) The data file is encrypted and sent to the cloud along with the complete access policy with is also padded with false attributes.

- 4) When a doctor wants to read the data file, he connects to the cloud to get this file. His attributes are sent to the trusted authority which will create the intermediate key.
- 5) This intermediate key is sent to the fog nodes which partially decrypt the ciphertext. This process also includes testing the partial access policy (see CP-ABE section)
- 6) After decryption, all fog nodes send the partial ciphertext to the doctor which decrypts the complete ciphertext using his private key.

9. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new collaborative approach based on CP-ABE. In our approach, we used the Fog nodes to reduce the bandwidth and to decrease the decryption cost by delegating the decryption process to the fog nodes. This allowed us to reduce the complexity (at the user level) to one exponentiation and multiplications operations instead of the pairing operations which are energy-intensive.

Our solution also preserves the privacy of the access policy so that the data owner attribute information is not disclosed. This is performed by introducing false attributes which are mixed with the real attributes.

As future work, we plan to evaluate our approach on real devices. Our work also improved by using more expressive access policies (i.e., policies with ANDs and ORs). Bandwidth can also be optimized by reducing the number of communications between the Fogs and user.

REFERENCES

- [1] Hany F. Atlam, Robert J. Walters, and Gary B. Wills. Fog computing and the internet of things: A review. *Big Data and Cognitive Computing*, 2(2), 2018.
- [2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334, Los Alamitos, CA, USA, may 2007. IEEE Computer Society.
- [3] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attributebased encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06*, pages 89–98, New York, NY, USA, 2006. ACM.
- [4] Zhibin Zhou and Dijiang Huang. Efficient and secure data storage operations for mobile cloud computing. In *Proceedings of the 8th International Conference on Network and Service Management, CNSM '12*, pages 37–45, Laxenburg, Austria, Austria, 2013. International Federation for Information Processing.
- [5] L. Touati, Y. Challal, and A. Bouabdallah. C-cp-abe: Cooperative ciphertext policy attribute-based encryption for the internet of things. In *2014 International Conference on Advanced Networking Distributed Systems and Applications*, pages 64–69, June 2014.
- [6] Kim Thuat Nguyen, Nouha Oualha, and Maryline Laurent. Securely outsourcing the ciphertext-policy attribute-based encryption. *World Wide Web*, 21(1):169–183, January 2018.
- [7] Kai Fan, Junxiong Wang, Xin Wang, Hui Li, and Yintang Yang. A secure and verifiable outsourced access control scheme in fog-cloud computing. *Sensors*, 17(7), 2017.

- [8] Jinmiao Wang and Bo Lang. An efficient and privacy preserving cp-abe scheme for internet-based collaboration. In CollaborateCom, 2017.
- [9] Sana Belguith, Nesrine Kaaniche, Maryline Laurent, Abderrazak Jemai, and Rabah Attia. Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot. Computer Networks, 133:141 – 156, 2018.

UPGRADING CLOUD INFRASTRUCTURE – CHALLENGES AND SOLUTIONS

Andrei Petrescu and Mihai Carabas

University POLITEHNICA of Bucharest, Splaiul Independentei 313, Bucharest,
Romania

ABSTRACT

In today's fast-moving world, advances in technology occur at an alarming rate. Keeping up is difficult, but mandatory, and we must find solutions that will make the process easy. Out of all these technologies, cloud computing is one that is evolving the quickest. We will explore the tools which will help us help us reach our goal and talk about the main subject of our paper, namely keeping up to date with the latest releases in OpenStack private cloud technology. We will also talk about the results and how we found the best solution for the context in which this paper lies.

KEYWORDS

cloud, openstack, cinder, nova, keystone, glance, heat

1. INTRODUCTION

The term “cloud computing” has been around since the 1990s, but the first depiction was observed in a 1977 drawing of a multi-network diagram that described connections between ARPANET, SATNET and Packet Radio net. It was only in 2006 when the term cloud computing [1] was used in the context that we know today, and the technology came to reality when Amazon launched Amazon Web Services and offered S3 (Simple Storage Service) for cloud storage, EC2 (Elastic Compute Cloud) for infrastructure and SQS (Simple Queue Service) for messaging queues.

The problem arising from the context which we described earlier is that institutions and companies have problems keeping up with new releases of said software and cannot benefit from the advances and fixes that they bring. The problem of keeping up with new releases arises from the fact that development is done in an agile way in Openstack community [9]: there is a new version once every six months. In the case of the Faculty of Computer Science and Information Technology, their OpenStack cluster has been stuck to the same release since 2015, when Openstack Liberty was developed. They did not upgrade the version of Openstack due to the fact that there was not present any clean methodology to do the upgrade without breaking any production services. Because there have been 5 releases since Liberty, the cluster suffers from a lack of features and stability which new features provide [15]. Another problem is that the Liberty release has reached its EOL (end of life) status, meaning that it will not receive any more updates. As time passes by, more problems will arise because the difference between the versions will grow, and it will become even harder to do upgrades without causing loss of data or increased downtime.

This paper proposes a methodology on how to upgrade Openstack from Liberty to Queens version without breaking anything in production. Thus, our main objective is to provide a way

Natarajan Meghanathan et al. (Eds) : CSEIT, CMLA, NeTCOM, CIoT, SPM, NCS, WiMoNe, Graph-hoc - 2019
pp. 73-82, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91306

to upgrade a cloud infrastructure based on OpenStack framework to the latest release, which at the time of writing this paper is Queens, from earlier releases.

The paper is structured as follows: Chapter 2 presents an overview of different cloud computing technologies, chapter 3 shows how to do configuration management which is a trivial step in managing cloud frameworks. Chapter 4 presents the proposed solution regarding the cloud infrastructure upgrade and chapter 5 is doing evaluation related to the services upgraded in Openstack.

2. CLOUD COMPUTING TECHNOLOGIES

Cloud computing [6] is a modern concept that enables users to abstract the hardware layer by using resources in a dynamic way and based on their needs, in a much faster way than using standard baremetal servers [16]. This concept is based around virtual machines, which can share physical resources and can be created and destroyed very fast. It all began with the concept of the GRID architecture [7] that describes the close coupling of computational resources to act like a single machine [2]. We can refer to a SMP (Symmetric Multiprocessor) as a grid of many colocated CPUs that do the same work, and to an MPP (Massively Parallel Processor) as a grid of SMPs interconnected by very fast busses. A cluster is a group of computers that share the same purpose [10]. A very well-known example is SETI@Home, that comprised of 400000 CPUs which belonged to computers all over the world, all serving the purpose of finding extra-terrestrial life [2]. In recent years, the term as-a-service has been coined, which describes the types of services that the cloud can offer [12]. The three big kinds of services, are as follows:

Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). Another type of service that appeared recently, where focus is shifted on the function that resources do, and not on the resources themselves is called Function as a Service (FaaS).

OpenStack is an IaaS [1] solution for private clouds, and one of the most popular among them, occupying second place, behind VMWare vSphere, and has been adopted by more than 1200 companies, including Best Buy, Comcast, PayPal, Walmart and Wells Fargo. It is an open source project that was initiated in 2010 and is now backed by more than 1300 active contributors.

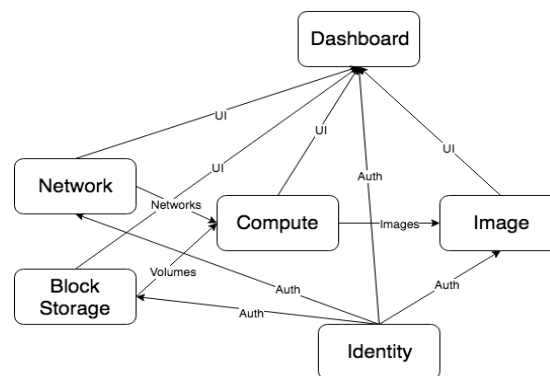


Figure 1. Flow of data in the OpenStack architecture

Its architecture is based mainly around controller and compute nodes but can also have optional nodes such as networking and storage nodes [8]. The controller nodes are the control plane of the cluster by holding the APIs of the services and performs authentication and scheduling of resources. They can also hold resources that are shared between components all over the cluster such as the database or the messaging queue [14]. The compute nodes are the ones that hold the

virtual machines and are usually in greater number than controller nodes and also can provide virtual machine live migration services [13]. The networking nodes are responsible for DHCP (Dynamic Host Configuration Protocol), VLANs (Virtual Local Area Networks), tunneling, routing and also for the flow of traffic in the cluster. The storage nodes are assigned the role of providing block storage volumes and are typically backed by LVM (Logical Volume Manager). There can also be nodes dedicated to generic object storage and image storage.

In this project we will focus mainly on a handful of important services but will briefly discuss others too. The most important services that are found in OpenStack are Keystone, Nova, Neutron, Cinder, Glance, Heat and Horizon.

Much of the current literature which describes the upgrade of OpenStack focuses on upgrading from release N to release N+1 by doing rolling updates with no downtime. This is not helpful in the context of the problem that this project will solve, because the difference of releases that will be covered will be 5, from Liberty to Queens.

3. CONFIGURATION MANAGEMENT

This section is focused on describing the notion of configuration management. The most important role of configuration management is to provide an easy and fast way to provision servers by using automation, thus eliminating human error. Before configuration management, the setup of servers was mainly done by hand, or by the use of bash scripts. The main problem with the old way of configuring servers is that it was error prone and it was not modular. Since the invention of configuration management, the term Infrastructure as Code has been brought up.

Infrastructure as code enables infrastructure to be treated as application code and be edited, reviewed and version controlled. System administrators could now track errors in their code and treat them as bugs, have repositories for their code and have different branches for testing and production. There are two types of Infrastructure as Code software on the market right now. The first type focuses on creating infrastructure, i.e. virtual machines, networks, IP allocation and so on. Examples of software that are specialized for these kinds of tasks are Terraform and Salt-cloud and work by accessing the APIs of the cloud platforms that they target. The second kind is the one that focuses on configuring servers by installing software, managing configuration files and ensuring that certain services are up and running. This is the type of Infrastructure as Code software that we will focus on and that we have used in my project.

Puppet is a configuration management software written in Ruby that is developed by Puppet Labs and is one of the first modern CM software, being launched in 2005. Puppet is based around a master-slave architecture, where the code resides on the master and the agents pull the code and run it locally.

Puppet code is based around the idea of modules, each serving a different purpose. In turn, modules are composed of three folders: files, manifests and templates. The files folder is used for static files, the templates are used for dynamically generated files, for example a template that generates a MySQL configuration file and the IP that it listens on is determined at run-time, and finally, there are manifests. Manifests are the core of Puppet and contain the actual code. They are made up of classes, each doing a specific task. Classes can build on other classes and modules can build on other modules. In OpenStack installation, there is a module named OpenStack, which builds on two other modules, OpenStack Controller and OpenStack Compute, and installs either one depending on the type of computer that is executing the code. OpenStack Controller builds on Puppet modules that install RabbitMQ, Memcached, Apache, MySQL, Keystone, and parts of Neutron and Nova that belong on a controller node. OpenStack

Compute builds on two Puppet modules that install the Neutron OpenVSwitch agent and the Nova Compute service.

4. PROPOSED SOLUTION AND IMPLEMENTATION DETAILS

As stated in the introduction, the purpose of this project is to upgrade the current version of OpenStack, which is Liberty, to the latest version, Queens. Because these two versions are 5 releases apart, the two main goals which we will achieve are to execute the upgrade fast and to preserve all the data. To be able to do this, we propose a solution where there are as few upgrades between services as possible. Traditional OpenStack upgrades are executed on every service that the cluster is running. In the case of the cluster that is running on the servers of the faculty, there are 7 services, 3 of which also run on different nodes than the controller node.

Cinder runs on two servers, the controller and the storage node, Nova runs on the controller and each compute node and Neutron runs on the controller, the network node and on each compute node. Suppose we have 1 storage node, 1 network node and 4 compute nodes. There would be 35 upgrades on the controller, 40 upgrades on the compute nodes, 5 on the storage node and 5 on the network node. In total, there would be 85 upgrades. The solution that we propose will introduce downtime, but it will preserve data and it will be relatively fast for getting through 5 releases.

Nodes other than the controller nodes, and the dashboard service, will be upgraded directly from Liberty to Queens because the data in the database is not modified by them. The data in the database is modified by the services which run on the controller node and synchronize the database on each release. Synchronization does two things, it either upgrades schemas or migrates data. Sometimes columns are renamed, or their type is changed, and not synchronizing the database would cause the new version of the service to not start at all. Database version are called migrations levels and they need to be sequential.

By doing upgrades this way, using the same scenario as before, we will have 18 upgrades on the controller, 1 on the storage node, 1 on the network node and 2 on each compute node, so 22 in total, compared to 85.

Another proposition is using Puppet for managing the upgrades, as it can speed up the upgrade process and assure that every execution of the code will produce the same results. Even if we have narrowed down the updates to 22, the upgrade still needs to be done with great care. Because of this, the Puppet classes must do the least amount of work and be run manually so as to make sure that errors have not occurred. There should be classes for each service upgrade and the code should be copied on each node, so that it can be run using “puppet apply”. The connection strings in the configuration files must also be changed so that the services can successfully connect to the database. The user has updated the API endpoints or create new endpoints if necessary, which is the case for Cinder and Nova, because of the Placement API that is introduced in the Ocata release. Lastly, and most importantly, before synchronizing the database in Ocata, the user must create a database named “nova_cell0”, map cell0 and create a cell named “cell1”. This is mandatory in Ocata, as Nova has switched to this new, more scalable, system of managing compute nodes.

Lastly, the organization of the Puppet modules must be done so they cover all the possible deployments of OpenStack cluster. Some examples of OpenStack deployments are presented in Figure 2.

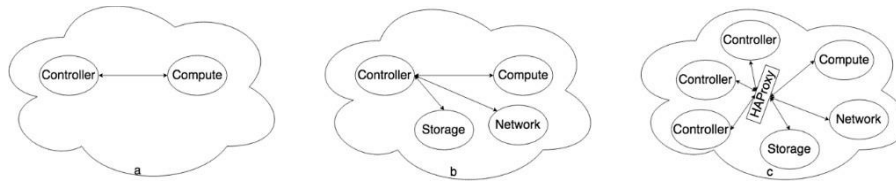


Figure 2. The figure presents three OpenStack deployment examples

In Figure 2a, OpenStack is installed on only two nodes, on what we call a proof of concept cluster. In Figure 2b, the Network and Storage nodes are separated from the Controller to provide better resource management. Figure 2c presented a high availability OpenStack deployment where there is more than one Controller node. Requests to the controller nodes go through a proxy for the reason of simplifying access and ensuring equal load on the nodes.

The first thing which we needed to create was an automated, repeatable and fast deployment mechanism in order start from scratch every time we have done a mistake that would cost us more time to fix than to start over again. To do this, we made use of the existing OpenStack cluster and created two virtual machines, with 4GB RAM each, which is the minimum recommended for a proof of concept deployment of OpenStack Queens. Because we needed to start from scratch every time, we made use of the rebuild feature that OpenStack Nova provides. This enabled the reinitialization of the instance with a fresh install of CentOS in a short period of time, so we could start over again and change the approach that we took to creating the up- grade process. Below, in table 1, the times for rebuilding and bootstrapping the Liberty deployment on the two nodes is presented.

Table 1. Deployment time benchmark.

Rebuild Time (s)	Deployment on controller (s)	Deployment on compute (s)
71	601.67	565.11

Information about upgrades between multiple releases is hard to find, so we chose to test the upgrade of each service from Liberty to Queens. This would often fail, and the main indication was that the synchronization of the database would fail.

Further, we will describe some interesting database errors that we have come across when upgrading Nova and the usual errors which appears when trying to upgrade the database schema of other services between releases which are too far apart. We will also present some packaging errors and some bugs that we have found in the OpenStack code.

OpenStack Mitaka introduces an important change in the Nova database, more specifically, in the compute_nodes table. There is one column that is added, named uuid, and because of this, it is important to re-register the compute nodes after upgrading. Figure 3 is outputted from a 2 select operations done on the compute_nodes table and describe how the uuid entry is filled after a compute node reconnects to the Nova API.

```

host: openstack-agent
ram_allocation_ratio: 0
cpu_allocation_ratio: 0
uuid: NULL
    
```

Figure 3. Difference in output before and after the node registered

If this step is skipped, when trying to upgrade to the Newton release, the upgrade script will output an error message, as shown in Figure 4, and will not continue until all the entries in the table with the value of NULL in the uuid column will be deleted.

```
error: There are still 1 unmigrated records in the compute_nodes table. Migration cannot continue until all records have been migrated.
```

Figure 4. Error outputted if nodes are not re-registered

When trying to upgrade to Ocata, it is also important to go through the Newton release, because there are database entries which need to be migrated into the nova_api database. Figure 5 shows the error message that the upgrade script outputs when data is not migrated.

```
ValidationError: Migration cannot continue until all these have been migrated to the api database. Please run 'nova-manage db online_migrations' on Newton code before continuing.
```

Figure 5. Ocata - migrations were not done beforehand in Newton

An example of the data migration talked about earlier is presented in Figure 6. It shows the output after the migration is done because, before that, the table was empty.

```

MariaDB [nova_api]> select * from flavors;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| created_at | updated_at | name | id | memory_mb | vcpus | swap | vcpu_weight | Flavorid | rxtx_factor | root_gb | ephemeral_gb | disabled | is_public |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2018-06-18 17:13:13 | NULL | m1.medium | 1 | 4096 | 2 | 0 | 0 | 3 | 1 | 1 | 40 | 0 | 1 |
| 2018-06-18 17:13:13 | NULL | m1.tiny | 2 | 512 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 2018-06-18 17:13:12 | NULL | m1.large | 3 | 8192 | 4 | 0 | 0 | 4 | 1 | 80 | 0 | 0 | 1 |
| 2018-06-18 17:13:13 | NULL | m1.xlarge | 4 | 16384 | 8 | 0 | 0 | 5 | 1 | 160 | 0 | 0 | 1 |
| 2018-06-18 17:13:13 | NULL | m1.small | 5 | 2048 | 1 | 0 | 0 | 2 | 1 | 20 | 0 | 0 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 6. Select operation after data has been migrated in the Newton release

Another important aspect when taking into consideration the upgrade to Ocata, is to create the nova_cell0 and the cell mappings, as they are mandatory from that release on. If these steps are not done before the synchronization database, the upgrade script will out an error as show as in Figure 7.

```
ValidationError: Cell mappings are not created, but required for Ocata. Please run nova-manage cell_v2 simple_cell_setup before continuing.
```

Figure 7. Error of the Ocata upgrade - cell mapping was not done beforehand

Other common errors are related to packages which the package manager does not upgrade automatically, and the services either fail to start or the database migration fails because of them. Below we describe one of them and the process that we went through to fix it and others which were related.

When trying to migrate Cinder from Liberty to Newton, because the package manager sometimes does not update all dependencies. This error can be resolved by upgrading the

python2-os-brick package. Other errors like these can be resolved the same. We looked at packages that were imported by the Python module and searched what version is installed by using the command “yum list installed” and then piping the output to grep.

Upgrading services too far will cause errors to be outputted by the management script. One example is shown in Figure 8 and checking the current migration level in the database is shown in Figure 9.

```
[root@openstack-master openstack_upgrade]# /bin/heat-manage db_sync
ERROR: "Database schema file with version 66 doesn't exist."
```

Figure 8. Error output if Heat is upgraded too far

```

MariaDB [heat]> select * from migrate_version;
+-----+-----+-----+
| repository_id | repository_path | version |
+-----+-----+-----+
| heat         | /usr/lib/python2.7/site-packages/heat/db/sqlalchemy/migrate_repo | 65 |
+-----+-----+-----+
    
```

Figure 9. Finding the current migration level of the database

The most interesting kind of errors were those where all the imported packages in the Python modules were up to date and changes in the code were needed. These occurred when upgrading Cinder to the Newton release or Nova to the Ocata release.

These errors were caused by small bugs in the api.py module from the “sqlalchemy” database upgrade packages. Because they were looking for a profiler group in the configuration files of both services, and because those did not exist, the synchronization of the database would fail. Fixing the errors was done by adding a check for profiler attribute in the CONF object.

Each step of the process is done automatically by the classes from the Puppet module that we have developed. To satisfy the constraint of modularity, where the kind of deployment does not matter, we have created 4 types of classes, those with “api” in the name are applied to the controller nodes, those with “comp” in the name are applied to the compute nodes, those with “net” in the name are applied to network nodes and those with “store” in the node are applied to the storage nodes. It does not matter which type is upgraded first, but it is important that all the types of nodes are running OpenStack Queens when starting all the services. The recommended order of upgrading for the controller nodes is presented in Figure 10. The other types of nodes can be safely upgraded directly from Liberty to Queens because they do not store data anywhere.

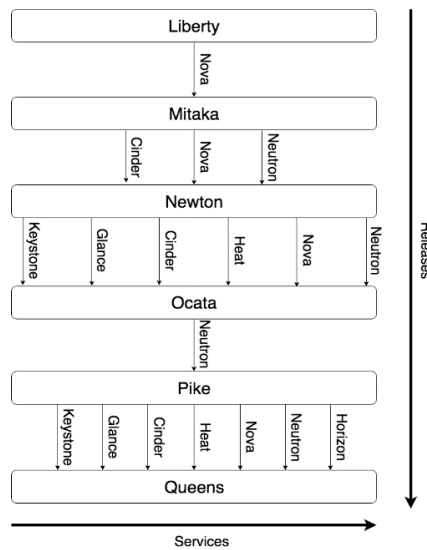


Figure 10. The figure presents the correct order of applying the classes

5. CLOUD INFRASTRUCTURE UPGRADE EVALUATION

In this section we will describe the tests which we have created to benchmark OpenStack after it was upgraded to Queens. We will focus on functional tests which will demonstrate that the cluster works as expected. To verify the functionality of the newly upgraded cluster, we decided to use 3 types of operations on 2 categories of resources, namely add, delete and edit on new and old resources. Table 2 shows what resources have been tested and the results.

Table 2. This table enumerates the types of test that we performed and the results

Name	Type	Add	Delete	Edit
Instances	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Images	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Key Pairs	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Networks	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Routers	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Security Groups	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Projects	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Users	Old	-	Yes	Yes
	New	Yes	Yes	Yes
Roles	Old	-	Yes	Yes
	New	Yes	Yes	Yes

6. CONCLUSIONS

In the beginning of the paper we have presented a brief introduction into the notion of cloud computing and have discussed about the context in which the subject of the paper lies, the problems which arise in this context, the objectives that the paper will meet, the proposed solution to meet these objectives, and an overview of the structure and the sections. The motivating factors which led us to choose this project were mainly of technical nature and had to do with advances which would benefit the students of the faculty in area such as artificial intelligence, machine learning and container orchestration. Moreover, there were personal reasons too that related to learning these new technologies which were used and providing an open source solution that the community can build upon and add new features.

To give an overview of the state of the art we provided a detailed overview of the technologies which form the subject of this paper. The first major technology discussed was cloud computing and how it came to be, as well as the types of services it provides. There are three important types of services which are discussed: Infrastructure as a Service, Platform as a Service and Software as

a Service. We discussed the main type of cloud provider for private and public clouds, OpenStack, which is the subject of this paper. We presented its architecture and the types of services that it is composed of, like the image service, the compute service, the block storage service, the identity service and the networking service. The second technology we have talked about is configuration management (Puppet) and how it is used to create infrastructure as code. We discussed about our proposed solution and the objectives that it should meet, namely that it should be faster than normal upgrades, simple to use, and most important of all, keep the data intact. It also presents in detail on what services should be upgraded to what release and how it should suite any type of OpenStack deployment. The last section discusses benchmarking by performing functional after the cluster was upgraded to ensure that the data was not corrupted and that old resources were still usable.

In conclusion, OpenStack will not stop here, and we will see new releases every year in the future, maybe at an even faster pace than today. To keep up with the changes in technology we decided to make our project available as open source on GitHub and will write blog posts on how to use it on our faculty's blog.

ACKNOWLEDGEMENT

The work has been funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

REFERENCES

- [1] Aniruddha S. Rumale, D.N.Chaudhari , „Cloud Computing: Infrastructure as a Service”, International Journal of Inventive Engineering and Sciences, vol. 1, no. 3, pp 1-7, 2013
- [2] Swarnpreet Singh and Tarun Jangwal , „Cost breakdown of Public Cloud Computing and Private Cloud Computing and Security Issues”, International Journal of Computer Sci-ence & Information Technology, vol. 4, no. 2, pp 17-31, 2012
- [3] Sumit Goyal, „Public vs Private vs Hybrid vs Community - Cloud Computing: A Criti-cal Review”, I.J. Computer Network and Information Security, pp 20-29, 2014
- [4] Borja Sotomayor, Ian Foster, Rubén S. Montero and Ignacio M. Llorente, „Virtual Infra-structure Management in Private and Hybrid Clouds”, 2009. [Online]. Available: https://www.researchgate.net/profile/Ian_Foster/publication/224587421_Virtual_Infrastructure_Management_in_Private_and_Hybrid_Clouds/links/00b49519a475a2ad95000000/Virtual-Infrastructure-Management-in-Private-and-Hybrid-Clouds.pdf
- [5] Tiago Rosado, Jorge Bernardino, „An Overview of Openstack Architecture”, IDEAS '14 Proceedings of the 18th International Database Engineering & Applications Symposium, pp 366-367, 2014
- [6] JoSEP, A.D., KAtz, R., KonWinSKi, A., Gunho, L.E.E., PAttERSon, D. and RABKin, A., 2010. A view of cloud computing. Communications of the ACM, 53(4).
- [7] Foster, I., Zhao, Y., Raicu, I. and Lu, S., 2008. Cloud computing and grid computing 360- degree compared. arXiv preprint arXiv:0901.0131.
- [8] Sefraoui, O., Aissaoui, M. and Eleuldj, M., 2012. OpenStack: toward an open-source solution for cloud computing. International Journal of Computer Applications, 55(3), pp.38-42.
- [9] Kumar, R., Gupta, N., Charu, S., Jain, K. and Jangir, S.K., 2014. Open source solution for cloud computing platform using OpenStack. International Journal of Computer Science and Mobile Computing, 3(5), pp.89-98.

- [10] Yadav, S., 2013. Comparative study on open source software for cloud computing platform: Eucalyptus, openstack and opennebula. *International Journal Of Engineering And Science*, 3(10), pp.51-54.
- [11] Corradi, A., Fanelli, M. and Foschini, L., 2014. VM consolidation: A real case based on OpenStack Cloud. *Future Generation Computer Systems*, 32, pp.118-127.
- [12] Merlino, G., Dautov, R., Distefano, S. and Bruneo, D., 2019. Enabling Workload Engineering in Edge, Fog, and Cloud Computing through OpenStack-based Middleware. *ACM Transactions on Internet Technology (TOIT)*, 19(2), p.28.
- [13] Hao, J., Ye, K. and Xu, C.Z., 2019, June. Live Migration of Virtual Machines in OpenStack: A Perspective from Reliability Evaluation. In *International Conference on Cloud Computing* (pp. 99-113). Springer, Cham.
- [14] Balmakhtar, M., Persson, C.J. and Rajagopal, A., Sprint Communications Co LP, 2019. Secure cloud computing framework. U.S. Patent Application 10/243,959.
- [15] Cotroneo, D., De Simone, L., Iannillo, A.K., Natella, R., Rosiello, S. and Bidokhti, N., 2019. Analyzing the Context of Bug-Fixing Changes in the OpenStack Cloud Computing Platform. arXiv preprint arXiv:1908.11297.
- [16] Moges, F.F. and Abebe, S.L., 2019. Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework. *Journal of Cloud Computing*, 8(1), p.2.

AN INTELLIGENT MOBILE APPLICATION TO MANAGE COLLEGE DATABASE AND RECOMMENDATION USING DATA MINING

Yixuan Qi¹, Qi Lu², Yu Su³ and Fangyan Zhang⁴

¹Valencia High School, Placentia, CA 92870

²Department of Social Science, University of California, Irvine,
Irvine, CA, 92697

³Department of Computer Science, California State Polytechnic University,
Pomona, CA, 91768

⁴ASML, San Jose, CA, 95131

ABSTRACT

College application is a critical and complicated task for high school students. Generally speaking, one student will submit an application to a number of universities or colleges. However, there is no proper software for them to organize their application-related information during the application process. This paper proposes an all-in-one system that can contain useful features that help students in their college application, such as compare his or her SAT/GPA, organize their rewards and activities, etc. This tool has been published in Google Play.

KEYWORDS

Android application, App development, Google Drive

1. INTRODUCTION

Based on the National Center for Education Statistics, about 19.9 million students are going to colleges in fall 2019. However, as helping tools, there are little resources can be used for college application. College application [1] has always been an endless and tedious process [2] and I have suffered a lot to organize my profile for the past 3 years. There is rarely any professional all-in-one application software [3] on the market.

In recent years more and more people are applying to colleges or universities. College application is the thing that most of the students must face and it is a long-term process that takes away people's time and energy. High school students will need to organize all their application-related information during their high school years. Tons of files will need to be distinguished into different categories. Maybe Google Drive [4] and such cloud storage can be used as the folder to store all student's information, a professionally designed software [5] will have a huge impact on the college applicants and high school students who wants to go to college. My project combines the three important features together avoiding additional time-wasting [6] and providing a professional storing folder [7] for a special purpose.

First of all, my project helps high school students who want to go to the college or university, as a data folder storing required personal materials and college information & requirements for all applicants. Moreover, this project contains the feature that every user can compare his or her

SAT/GPA/etc. [8]. with those from previous years. I am trying to simplify the college application process by providing a tool that is easy to access and contains an all-in-one system, with college Info & requirement comparison [9] & personal organizer [10], for helping college applicants.

There are many websites for people to compare their grades and test scores and such College Board (Figure 1) [11] has its own college information page or google drive can be used as an application folder.

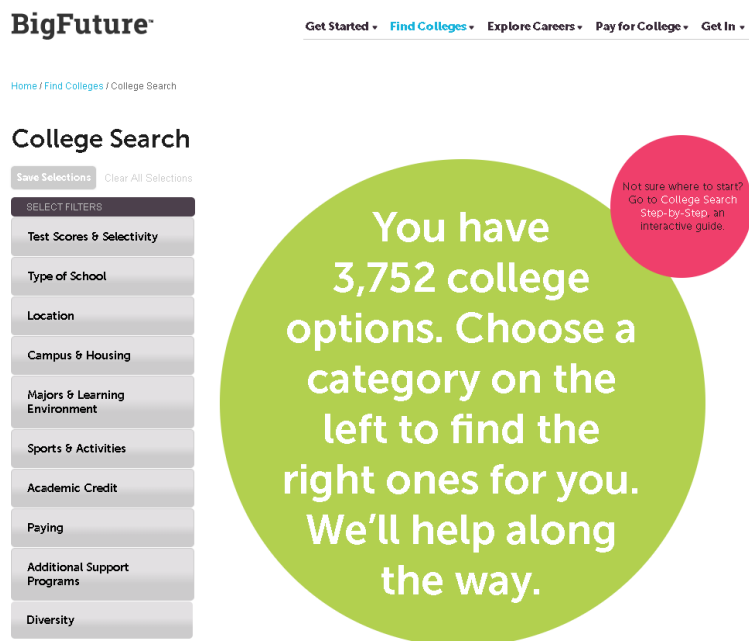


Figure1: CollegeBoardBigFuture

However, these tools/pages are usually not very easy to use or not professional enough. Sometimes people have to spend hours even days to organize some random files or projects and later they spend even more time finding out the location. There is also some information needed for college application but there is nowhere to keep them.

Websites like USNews (Figure 2) [12] has the most professional information and rankings of most every college and university people are looking for. However, there is usually nowhere to save the information and even there is the progress is very complicated. The information sometimes lacks the accessibility [13] in which makes people's lives much harder. All they need as a college applicant is to manage their time properly but spend hours opening the tabs and pages is really a waste of time.

My method is to create an all-in-one system in which all features are included, and users can use them anytime anywhere. For the information-saver problem, my app has special places for users to input everything needed. For example, they can input all their rewards and activities.

The first 2 sections talk about the significance of college application and the reason that my project is indispensable. Section 3 talks about the features and functions. Section 4 and 5 mainly discuss the benefits and advantages my project has.

BEST COLLEGES
USNews
RANKINGS

Emory University ☆
201 Dowman Drive, Atlanta, GA 30322 | (404) 727-6123
#21 in National Universities
Overall Score 79/100

2019 QUICK STATS	
Tuition and Fees	\$53,804 (2019-20)
Room and Board	\$14,972 (2019-20)
Total Enrollment	14,459
Application Deadline	Jan. 1

Overview Rankings User Reviews Questions & Answers Applying Cost & Aid Academics Student Life Services Safety More ▾

Overview of Emory University

Emory University is a private institution that was founded in 1836. It has a total undergraduate enrollment of 7,086, its setting is city, and the campus size is 631 acres. It utilizes a semester-based academic calendar. Emory University's ranking in the 2020 edition of Best Colleges is National Universities, #21. Its tuition and fees are \$53,804 (2019-20).

Emory University, located near downtown Atlanta, is divided into nine schools and colleges, four of which serve undergraduate and graduate students. Emory's graduate programs include the highly ranked *Goizueta Business School*, *School of Law* and *School of Medicine*. First- and second-year students are required to live on campus, but a majority of students remain on campus all four years. The Student Programming Council organizes events and performances throughout the year, including from well-known entertainers and musicians. A popular student organization is

MORE FROM THIS SCHOOL

- Colleges
- Global Universities
- Graduate Schools

My Fit Custom College Ranking
Does this school fit your college needs? Receive a personalized ranking provided by U.S. News College Compass and find out. **TRY IT NOW** »

Figure2: USNewsCollegeSearch

2. MOTIVATION

I built this app because based on my situation, I think this app could really help those high school students who are going to college or university. I think such an app could save time for them so they can focus on some meaningful tasks instead of wasting time here. The challenge I faced was that I had to decide what function should the app have. The main point of this app is to create efficiency [14], so I had to choose what functions were unnecessary. I interviewed with some students in my high school and made the final decision.

3. SOLUTION

First of all, my app starts with a sign up/login page (Figure 3a), where users can create their own accounts the first time, they use it and have their own private online storage [15] parts. After logged in users will go to the main page (Figure 3b) where the four major functions are implemented. Users can go to the specific pages editing their scores, activities, college wish list, and important dates.

If users choose to go to the score page (Figure 5a), they can see all the scores they entered and when then click the add score button they will jump to the other page to add a new score or update the latest scores they have (Figure 5b). After adding/updating the scores, users click the add button and the scores are now added/updated to the cloud storage. New scores are added to the page and undated scores are now storing the latest information (Figure 5c).

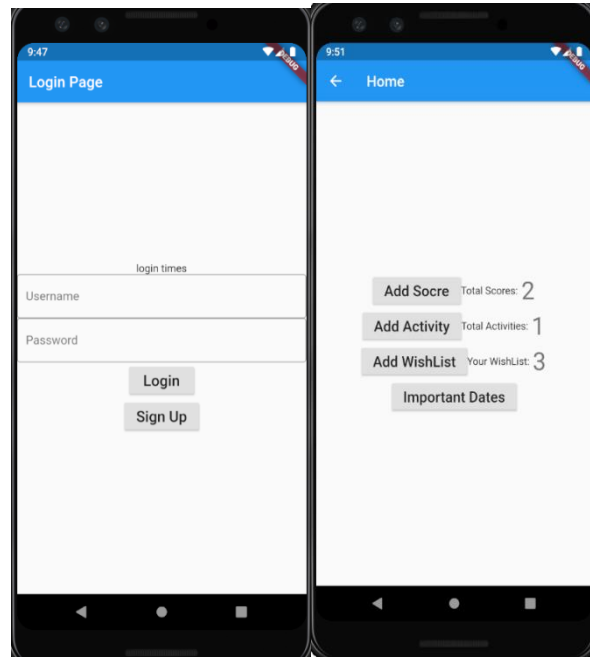


Figure 3: (a) Login Page (b) Main Page

```

RaisedButton(
  onPressed: () {
    server.signIn(usernameController.text, passwordController.text)
      .then((uid) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => HomePage(title: 'Home')), // MaterialPageRoute
          );
      }).catchError((e) {
        print("failed to login");
      });
  },
  child: const Text(
    'Login',
    style: TextStyle(fontSize: 20)
  ), // Text
), // RaisedButton

RaisedButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => SignUpPage(title: 'SignUp')),
    );
  },
  child: const Text(
    'Sign Up',
    style: TextStyle(fontSize: 20)
  ), // Text
), // RaisedButton

```

Figure 4: Code for the App

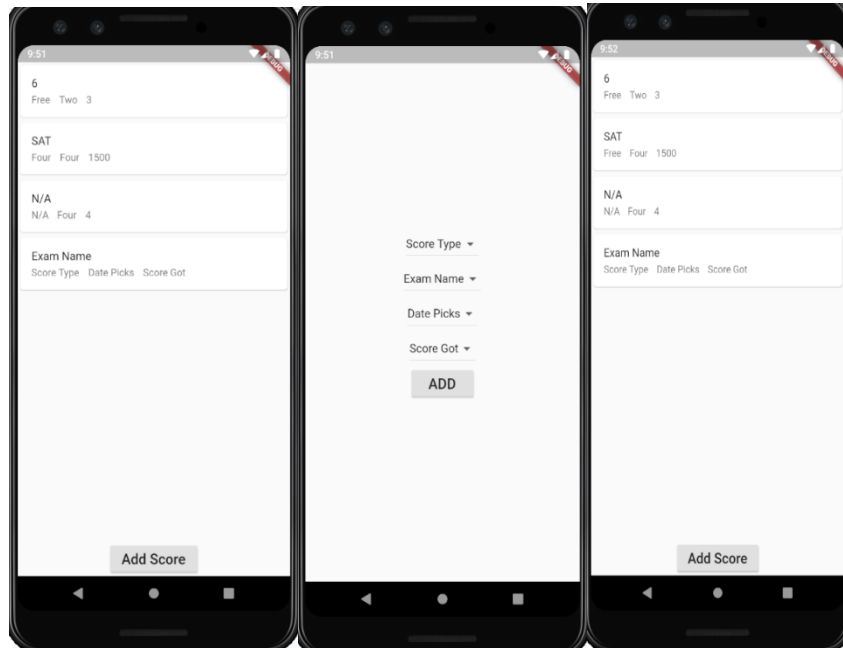


Figure 5: (a) ScoreEntered Page (b) Enter Score Page (c) Back to ScoreEntered Page with New Info

Users could also go to the activity/important Date (Figure 6a and 6c) page by clicking add Activity or add Important Date (Figure 6b) button, the functions are basically the same as the score page where people can edit their activities they have done or important dates they have.

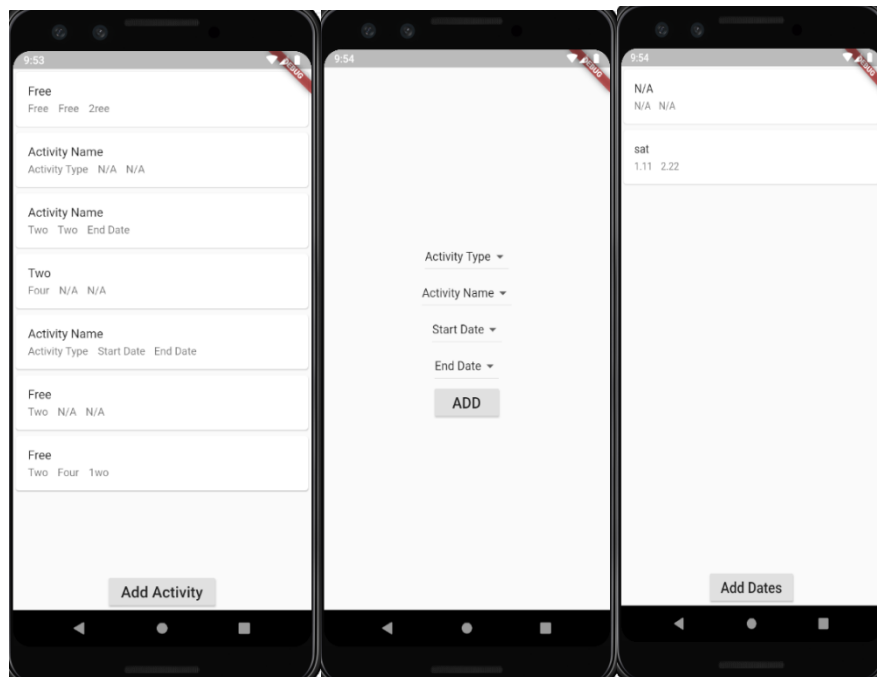


Figure 6: (a) ActivityEntered Page (b)Enter ActivityPage (c) ImportantDate Page

By clicking the Add Wishlist button (7a), users will jump to the college list page and they can find most of the information they need related to the application.

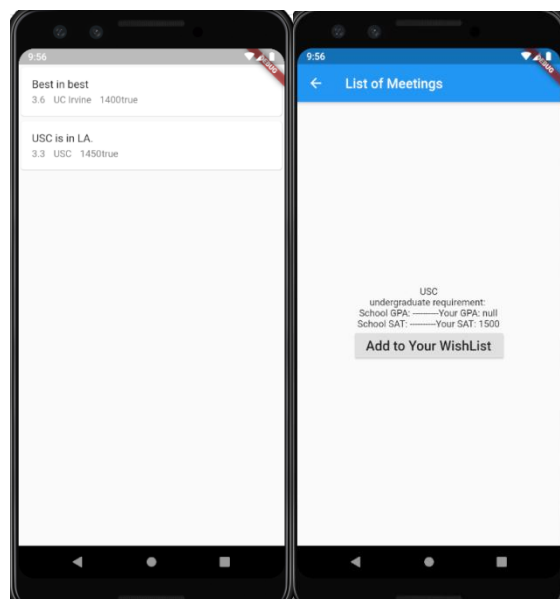


Figure 7: (a) Wishlist Page (b) Add College Wishlist Page

Users can click the schools they wish to apply, and the app will jump to the Add College Wishlist page (Figure 7b). The page will show basic information about the college/university and there are two comparisons between GPA and sat/act scores. The app will automatically take the scores users entered & saved to compare to the average applicant's scores. Users can then click AddWishList to add the college to their wish list and the page will jump back to the College Wishlist page. The college/university users selected will be marked with some color/icon presenting the specialty.

4. RELATED WORK

Due to the function google drive has, which is an online folder, most college applicants/high school students use Google Drive for storing their related activities, rewards, etc. However, a general folder means that those students have to find the college information elsewhere and after months or even years there is so much information, whether related or not, have to be organized. College Board has its own college wish list where students can select their dream schools and check the information and application requirements. However, students do not have College Board accounts until they are taking their first AP test or SAT test. Moreover, the College Board has little accessibility that barely anyone uses it on the phone or the only thing they do is check their scores.

5. CONCLUSION

In this project, we proposed an all-in-one system that helps high school students in their application. This tool is a mobile app [16] that has been developed to manage application materials. The system is able to alleviate pressure and effectively reduce applicants' duplicate tasks [17]. The result shows that this tool can help high school students in their college application.

As for the future work, we will investigate thoroughly high school student's application process to make sure that the system is updated and cover cases as many as possible. We also would like to explore and make it more complete.

In addition, one limitation related to the app is that it does not have enough users in test. we plan to add more features to the system in the next version and follow high school students' applications from the beginning to the end.

REFERENCES

- [1] Roderick, Melissa, Vanessa Coca, and Jenny Nagaoka. "Potholes on the road to college: High school effects in shaping urban students' participation in college application, four-year college enrollment, and college match." *Sociology of Education* 84, no. 3 (2011): 178-211.
- [2] Robinson, Karen Jeong, and JosipaRoksa. "Counselors, information, and high school college-going culture: Inequalities in the college application process." *Research in Higher Education* 57, no. 7 (2016): 845-868.
- [3] Hallar, James H., and Wai Lim Chan. "All-in-one information handling system." U.S. Patent Application 29/468,397 filed April 29, 2014.
- [4] Quick, Darren, and Kim-Kwang Raymond Choo. "Google Drive: Forensic analysis of data remnants." *Journal of Network and Computer Applications* 40 (2014): 179-193.
- [5] Quinn, James Brian, Philip Anderson, and Sydney Finkelstein. "Managing professional intellect: making the most of the best." *The strategic Management of Intellectual capital* 87100 (1998).
- [6] Haycock, Kati, and Stephanie Robinson. "Time-Wasting Workshops?" *Journal of Staff Development* 22, no. 2 (2001): 16-18.
- [7] Behrens, Dietmar, Waldemar Hoog, and Rudolf Zimmermann. "Device for storing a data diskette in a file folder." U.S. Patent 4,884,691 issued December 5, 1989.
- [8] Nofle, Erik E., and Richard W. Robins. "Personality predictors of academic outcomes: big five correlates of GPA and SAT scores." *Journal of personality and social psychology* 93, no. 1 (2007): 116.
- [9] Echerer, Scott J., and Stephen R. McNeill. "Radiographic image enhancement comparison and storage requirement reduction system." U.S. Patent 5,740,267 issued April 14, 1998.
- [10] Randall, Stephen. "Electronic personal organizer." U.S. Patent 5,237,651 issued August 17, 1993.
- [11] Ramist, Leonard. "College Student Attrition and Retention. College Board Report No. 81-1." (1981).
- [12] Weaver, David Hugh, G. Cleveland Wilhoit, and Lori A. Bergen. *The American journalist: A portrait of US news people and their work*. Indiana University Press, 1991.
- [13] Clemons, Eric K., and Steven O. Kimbrough. "Information systems, telecommunications, and their effects on industrial organization." (1986).
- [14] Ross, Stephen A. "Options and efficiency." *The Quarterly Journal of Economics* 90, no. 1 (1976): 75-89.
- [15] Gonzalez, José Luis, Jesus Carretero Perez, Victor Sosa-Sosa, Juan F. Rodriguez Cardoso, and Ricardo Marcelin-Jimenez. "An approach for constructing private storage services as a unified fault-tolerant system." *Journal of Systems and Software* 86, no. 7 (2013): 1907-1922.
- [16] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 15-24. IEEE, 2013.
- [17] Da-You, LI Jia-Fei LIU, and Y. A. N. G. Bo. "Process Mining: An Extended α -Algorithm to Discovery Duplicate Tasks [J]." *Chinese Journal of Computers* 8 (2007).

AUTOMATED GENERATION OF COMPUTER GRADED UNIT TESTING-BASED PROGRAMMING ASSESSMENTS FOR EDUCATION

Sébastien Combéfis^{1,2} and Guillaume de Moffarts²

¹ECAM Brussels Engineering School, Brussels, Belgium

²Computer Science and IT in Education ASBL, Louvain-la-Neuve, Belgium

ABSTRACT

Automatic assessment of code, in particular to support education, is an important feature included in several Learning Management Systems (LMS), at least to some extent. Several kinds of assessments can be designed, such as exercises asking to “fill the following code”, “write a function that”, or “correct the bug in the following program”, for example. One difficulty for instructors is to create such programming exercises, in particular when they are somewhat complex. Indeed, instructors need to write the statement of the exercise, think about the solution and provide all the additional information necessary to the platform to grade the assessment. Another difficulty occurs when instructors want to use their exercises on another LMS platform. Since there is no standard way to define and describe a coding exercise yet, instructors have to re-encode their exercises into the other LMS. This paper presents a tool that can automatically generate programming exercises, from one single and unique description, and that can be solved in several programming languages. The generated exercises can be automatically graded by the same platform, providing intelligent feedback to its users to support their learning. This paper focuses on and details unit testing-based exercises and provides insights into new kinds of exercises that could be generated by the platform in the future, with some additional developments.

KEYWORDS

Code Grader, Programming Assessment, Code Exercise Generation, Computer Science Education

1. INTRODUCTION

Being able to automatically grade code produced by learners, and in particular students in schools and universities, is a very demanded feature for Learning Management Platforms (LMS) [1]. In particular, professors in charge of programming courses need to assess the programming skills of their students. It is of course also the case for other courses that may require some programming, such as data mining or natural language processing courses, for example. The main issue is that this assessment cannot be done manually, especially if there are a large number of students [2-3]. Another situation, where automatic code assessment is mandatory, is Massive Open Online Courses (MOOCs), for which students are spread all over the world and are even more numerous [4-5]. Of course, the automatic grading of code must be more advanced than just assessing whether the code compiles and produces the correct result for some test cases. It should provide useful feedback to the learners. It is even more important when the number of students is large or in the case of MOOCs for which learners have a more limited access to the professors to get more individualised feedbacks.

This paper proposes a tool to generate coding exercises that can be solved in several programming languages and that can be automatically graded. Exercises are generated from a single language-agnostic configuration file. The same configuration can therefore be used to

generate several instances of the same exercise for different programming languages. Feedbacks generated by the tool, and provided to the learners, are designed to help the learners to identify and understand their faults. They are also more suited for education and designed to support their learning. The first prototype of the tool [6] has been used to support a university course, at the Université catholique de Louvain (UCLouvain), introducing students to programming concepts and paradigms, as well as for a MOOC on the same topic [4]. The more recently rewritten version adds generic unit testing-based exercises [7]. It has been used for a second bachelor course about Python programming at the ECAM Brussels Engineering School, a higher education institution for future engineers. Finally, the last version of the tool, presented in this paper, supports automatic generation of unit testing-based exercises. It is currently tested with EDITx, a private company that organises IT challenges targeted to IT students and IT professionals, all around Europe.

1.1. Motivation

A lot of tools that can automatically grade codes do exist. They can generally be split in three categories: (a) code grading for programming competitions (online or onsite), (b) code evaluation for test-driven development and (c) code grading for education. For competitions, it is important to be able to guarantee the same execution environment and conditions for all the code evaluations. The main reason being that code evaluations are used to establish the ranking and to offer prizes to the participants. For example, it should be possible to impose time and memory limits that cannot be exceeded during the execution of participants' code submissions. Those graders must also be very robust to hold on during the whole competition, and must guarantee code and grading traceability in case of complaints [8]. For development, programs are typically tested to check whether their code is functionally correct regarding the executed test cases, following the Test-Driven Development (TDD) approach. For such assessments, time and memory constraints are less useful, but defining and controlling the test environment is also important. It should also be important to test the same code under different situations, for example to evaluate some fault tolerance levels. Finally, when it comes to assess code for educational purposes, several additional requirements arise. First, the feedback provided to learners must support their learning and cannot be limited to the classical "pass/fail" verdict of standard graders. The feedbacks must help learners to understand their faults and to make progress. Then, graders for education must support a larger number of different execution environments and programming languages than competition or development graders, that are often more specific. Finally, learner's code must be executed in a safe environment, for example isolated in sandboxes, because learners may produce wrong or dangerous code, whether it is voluntary or not.

All these observations led to the development of Pythia, a platform that combines requirements from the three categories of graders presented above. This platform has been designed to support education and, in particular, the teaching and learning of programming [6-7]. The main motivation that gave birth to the Pythia platform is to propose a tool on which several kinds of programming exercises can be automatically graded. It must also be flexible enough so that codes produced by the learners can be thoroughly analysed with existing tools. Therefore, the Pythia platform can support various assessments based on several criterions (functional correctness, code quality, execution performance, memory consumption, etc.). Finally, the platform should allow instructors to easily produce exercises following existing templates, or to build their own

exercises specifically tailored for their students. From the competition graders, Pythia took two ideas: isolated sandboxes to safely execute code and possibility to impose constraints (such as time and memory limits). From the “TDD graders”, Pythia took the idea of the systematic way to test codes against test suites. Finally, from education graders, Pythia took the idea of working on tailored “intelligent” feedbacks that support learning.

1.2. Related Work

As detailed above, many code graders have been developed, but most of them are either competition graders or specific ones only being able to handle certain kinds of exercises [9]. Several reviews have been conducted and interested reader can refer to them [10-13]. Among those graders, some follows the “TDD grader” philosophy and are based on tests [13]. When graders are to be used for educational purposes, reviews agree that feedback is important and that good feedback helps the learners and support their learning [14-15]. Finally, concerning automatic generation of programming exercises, only some solutions have been developed, but it is important in particular in the case of large classes to be able to easily diversify the number of available exercises [16-17].

The remainder of the paper is structured as follows. Section 2 presents a global overview of the architecture of Pythia. Then, Section 3 presents how to define an exercise and how it will be generated. Finally, Section 4 concludes the paper with some discussions and future works.

2. PYTHIA PLATFORM ARCHITECTURE

Pythia is a distributed application with several components. It has mainly been developed with the Go programming language. It uses UML virtual machines to execute code in a safe and controlled environment. The details of the architecture of the Pythia platform not being the purpose of this paper, the interested reader can refer to [6], or can directly delve into its code available here: <https://github.com/pythia-project>, to get a better understanding of it. Figure 1 shows a global overview of the architecture of the Pythia platform. The client interacts with the platform through an API server. This latter is connected to the Pythia backend, which manages the code execution within virtual machines (VM). Tasks to be executed and environments in which tasks can be executed are available to the backend and API server. They are in fact SquashFS read-only file systems stored on disk as files (TaskDB and EnvDB).

Two scenario examples are illustrated on Figure 1:

1. An instructor can call a specific route on the API server to create a new task. The task generator component will create it and store it in the TaskDB.
2. A learner can call a specific route on the API server to execute a task. The submission grader component will execute the task with the submission of the learner on the backend and return the generated feedback to the learner.

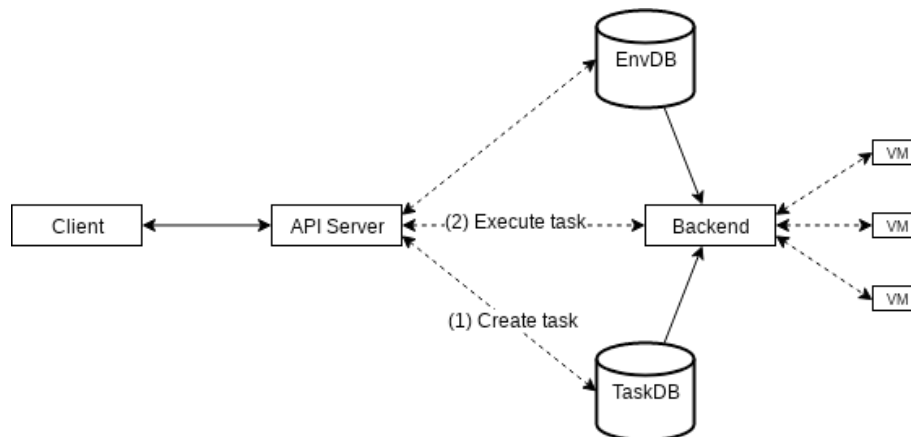


Figure 1. The Pythia platform is a distributed application with a backend managing VMs. It can be accessed through an API server that also manages tasks and execution environments.

2.1. Submission Grader

The submission grader is the component in charge of gathering the submission of the learner for a specific task and to evaluate it. The POST `/api/execute` route of the API server takes two parameters: the unique identifier of the task to execute and an input, which is a string containing the submission of the learner. The API server responds with three elements: the unique identifier of the task that has been executed, the status of the execution by the Pythia backend (success, timeout, overflow, etc.) and the output produced by the execution of the task (which contains among others the feedback). Depending on the kind of exercise, the input provided to the Pythia backend and the output produced by the execution of the task can be structured following a specified format. The Pythia platform does not impose anything on input and output. They just have to be strings, with a limitation on the number of characters for the output.

For example, Figure 2 shows the input and the output produced by the execution of a unit testing-based exercise where the student has to write the body of a function that computes the subtraction of its two arguments. The input should be a JSON object with two keys, one with a unique submission ID and one with the set of pieces of submitted code. For this particular exercise, there was only one field to fill out, named `f1`. The produced output contains the unique submission ID, the status of the execution of the tests (success, failed), and some feedback information. In this case, the feedback contains four elements:

- a score: 0.14285715,
- some statistics about the tests: 2 succeeded tests on a test suite with 14 tests,
- an example of inputs for which a test failed: for input (10,5), the expected answer is 5 (that is, $10 - 5$) and the answer computed by the learner's code is 10, and finally
- a message to help the learner find his/her fault: "Have you subtracted the 2nd parameter?"

The score and the statistics help the learner to evaluate how far from the completion of the exercise he/she is. The goal is to reach a score of 1, that is, to succeed all the tests from the test suite. The learner can also evaluate his/her own progress between submissions for the same exercise thanks to those statistics. Thanks to the example of inputs for which a test failed, the learner can trace his/her code execution to understand why it produced a wrong result. The learner can also check his/her corrected code before submitted it again, thanks to the provided expected answer. Finally, the message associated to the example of input should help the learner

to find his/her fault. Since this message is more intuitive and related to the statement of the exercise to solve, it should encourage the learner to think about his/her solution, and not to try to change the code just to pass the failed test.

The handling of this specific input and the generation of this specific output are managed by code embedded in this particular task. In the Pythia platform, a task is in fact just a bunch of code that is executed in a safe environment, namely the UML virtual machine, taking a string as input and producing a string as output. An instructor can therefore create any kind of exercise, as long as he/she is able to write a code to parse the provided input, to evaluate the learner submission and to produce an output. He/she also has to define precise specifications for the input provided to his/her task and the generated output. Since the execution takes place inside a Linux virtual machine, the instructor can use any existing tool running on Linux to write a task. The only flip side of such flexibility is that creating a task can be very time-consuming and limited to only some instructors that have high programming skills and that understand the internal working of Pythia environments and tasks.

```

===INPUT EXAMPLE===
{
  "tid": "sub",
  "input": "{\"tid\": \"s001\", \"fields\": {\"f1\": \"return a\"}}"
}

===OUTPUT EXAMPLE===
{
  "tid": "s001",
  "status": "failed",
  "feedback": {
    "example": {
      "input": "(10,5)",
      "expected": "5",
      "actual": "10"
    },
    "message": "Have you subtracted the 2nd parameter?",
    "stats": {
      "succeeded": 2,
      "total": 14
    },
    "score": 0.14285715
  }
}

```

Figure 2. The execution of a unit testing-based task requires specific input information with the pieces of submitted code and produces a specific output with “intelligent” feedback information.

2.2. Task Generator

The task generator is a component in charge of automating the creation of tasks based on predefined templates. It can be used to ease the creation of exercises on the Pythia platform for instructors. For that, task templates must be defined, that is, a highly configurable generic program with placeholders must be designed as a task. Unit testing-based tasks [7] are structured following four processes as shown on Figure 3. The execution goes as follows:

1. The input of the learner is pre-processed, and used to fill a template code to produce the student code. This first step also initialises several files and directories. For example, it saves the task ID (tid) in a text file so that it can be used at the end of the task execution to generate the output of the task.
2. A test suite is then automatically generated based on the test configuration of the task contained in the test.json file. This file contains a set of predefined tests and

configuration information to generate random tests. The test suite is stored with the CSV format in the data.csv file.

3. The student code is then executed for each test of the test suite and the results of each execution are stored in the data.res text file. Each line of this file contains the verdict of the execution (checked, exception, etc.) with an associated value (the produced result, the description of the exception, etc.). Student code is executed in an unprivileged mode inside the virtual machine, so that it cannot access the correct solution or view some configuration files, for example.
4. Finally, the correct solution stored in the solution.json file is fed in the template code to produce the teacher code, which is executed to produce the correct solutions for the generated test suite. Solutions are stored in the solution.res text file, each line containing the correct answer for each test. Then, the feedback is generated, comparing the correct answers with the ones produced by the learner. The test.json file is again used, to get information about the predefined tests and customised feedback messages.

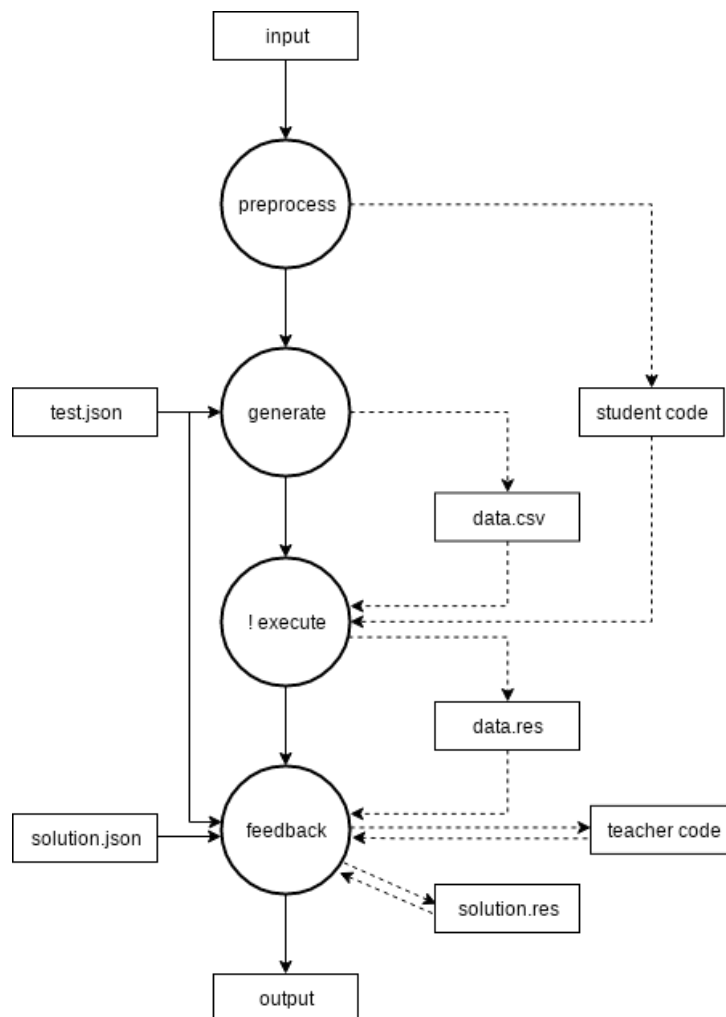


Figure 3. The structure of a unit testing-based task is composed of four main processes, namely the pre-processing, the tests suite generation, the code execution and the feedback generation.

Following this general structure for a unit testing-based task, it is possible to automatically generate exercises just providing some configuration information, described in the following section. Moreover, the only parts that are language-dependent are the execution of the student and the teacher code, all the rest being language-agnostic. To ease the implementation of unit testing-based tasks, the language-agnostic parts have been implemented as an independent tool written with the Go programming language, so that to be efficient. The language-dependent parts are implemented as libraries written in the target language for the exercise. For now, those libraries have only been written for the Python and Java programming languages.

3. ASSESSMENT STRUCTURE

To create a new exercise following the unit testing-based task template, an instructor has just to provide some basic configuration information structured as one JSON file, such as the one shown on Figure 4. The configuration consists in three distinct parts: (a) the specification, (b) the tests and (3) the solution. Except for the correct solution, all the other parts are language-agnostic and analysed either by the Go tool or by the language-dependent library. This task example asks the learner to write the body of a function `sub` that takes two parameters `a` and `b`, and that should return their subtraction, that is, `a - b`.

```
{
  "spec": {
    "name": "sub",
    "args": [
      {
        "name": "a",
        "type": "int"
      },
      {
        "name": "b",
        "type": "int"
      }
    ],
    "return": "int"
  },
  "test": {
    "predefined": [
      {
        "data": "(10, 5)",
        "feedback": {
          "10": "Have you subtracted the 2nd parameter?"
        }
      },
      {
        "data": "(7, 15)"
      },
      {
        "data": "(-1, 2)",
        "feedback": {
          "**": "Have you considered negative parameters?"
        }
      },
      {
        "data": "(12, 0)"
      }
    ],
    "random": {
      "n": 10,
      "args": [
        "int(-20,20)",
        "int(-20,20)"
      ]
    }
  },
  "solution": {
    "fl": "return a - b"
  }
}
```

Figure 4. A unit testing-based task can be generated from a configuration file containing information about the specifications of the function to write, information about the predefined and random tests to be executed and finally one correct solution for the task.

The configuration file consists of three parts:

- The specification part (spec) is used to generate the code templates from which the student and teacher codes will be generated thanks to the input submission from the learner and the correct solution from the instructor. It contains all the information related to the signature of the function that the learner has to implement for the task.
- The tests part (test) contains predefined tests that have to be run and information and constraints used to generate random tests. It also contains information about customised feedback message that can be produced to help the learner if he/she fails the test.
- Finally, the solution part (solution) contains one possible solution for the task. It consists of chunks of code that are used to generate the teacher solution that is executed to get the correct answers for the test suite.

The POST /api/tasks route of the API server takes several parameters among which the type of the task to create can be specified (unit-testing for unit testing-based tasks) along with the configuration (such as described by Figure 4) and the programming language. The task generator then builds a Pythia task with all this information, using the language-agnostic code for the pre-process, generate and feedback components and the language-specific code for the execute component. Thanks to this feature, an instructor can generate a coding exercise without having to write any line of code. A user interface can be designed to help instructor design such exercise visually, wrapping the creation of the JSON configuration file and the call to the API server. An experiment conducted by the EDITx private company is currently underway, asking higher education professors in charge of introductory programming courses at the bachelor level to write unit testing-based exercises thanks to the proposed platform.

4. CONCLUSIONS

The tool presented in this paper is the result of further development of the version of [7], which now has the ability to automatically generate unit testing-based exercises that can be automatically graded. An instructor willing to design an exercise does not have to write any lines of code, except to provide one correct solution for the exercise. The presented tool combines advantages from competition, TDD and education graders so that to be used for education and learning purpose. It can also generate “intelligent” feedbacks to support learning, providing the learner with hints about his/her faults. The automatic generation of exercises process has been designed to be easy which should encourage instructors to create more exercises for their learners. It should also encourage easier sharing between educators.

Of course, the main strength of the Pythia platform being its high flexibility, future developments of the platform include the addition of new kinds of exercises, with the automatic grading and the automatic generation parts. Writing a task for the platform is not easy, but thinking about a generic kind of task, from which instances can be easily created, without having to write any line of code is even less easy but way more interesting. Some insights about how to include input-output tasks to the Pythia platform, with the automatic grading and generation parts have already been found. The next feature will be the addition of those kind of exercises, where the instructor only provide a statement along with a set of string inputs with the corresponding expected string output. For such exercises, the instructor will no longer have to provide any line of codes to design a new task, since he/she will not even have to provide any correct solution.

The platform is currently being used for several courses at the ECAM Brussels Engineering School and on the IT challenges platform of the EDITx private company. Informal evaluations from usage of previous versions of the platform already showed that the platform does bring

useful help to learners. Future work includes a more rigorous evaluation of the platform and, in particular, should analyse the experiments in progress. Also, research has to be conducted to formally measure if the produced feedback information does indeed improve the learning performance of learners. It should also evaluate if the exercise creation process is easy and convenient enough for instructors.

REFERENCES

- [1] Pieterse, Vreda (2013). "Automated Assessment of Programming Assignments", in Proceedings of the 3rd Computer Science Education Research Conference (CSERC 2013), pp45-56.
- [2] Higgins, Colin A., Geoffrey Gray, Pavlos Symeonidis & Athanasios, Tsintsifas (2005). "Automated Assessment and Experiences of Teaching Programming", Journal on Educational Resources in Computing (JERIC), Vol. 5, No. 3, Art. 5.
- [3] Cheang, Brenda, Kurnia, Andy, Lim, Andrew, & Oon, Wee-Chong (2003). "On automated grading of programming assignments in an academic institution". Computers & Education, Vol. 41, No. 2, pp.121-131.
- [4] Combéfis, Sébastien, Bibal, Adrien & Van Roy, Peter (2014). "Recasting a Traditional Course into a MOOC by Means of a SPOC", in Proceedings of the European MOOCs Stakeholders Summit 2014 (EMOOCs 2014), pp.205-208.
- [5] Staubitz, Thomas, Hauke, Klement, Jan Renz, Ralf Teusner & Christoph Meinel (2015). "Towards Practical Programming Exercises and Automated Assessment in Massive Open Online Courses", in Proceedings of 2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE 2015), pp.23-30.
- [6] Combéfis, Sébastien & le Clément de Saint-Macq, Vianney (2012). "Teaching Programming and Algorithm Design with Pythia, a Web-Based Learning Platform", Olympiads in Informatics, Vol. 6, pp31-43.
- [7] Combéfis, Sébastien & Paques, Alexis (2015). "Pythia Reloaded: An Intelligent Unit Testing- Based Code Grader for Education", in Proceedings of the 1st Int'l Code Hunt Workshop on Educational Software Engineering (CHESE 2015), pp5-8.
- [8] Tochev, Tocho, & Bogdanov, Tsvetan. (2010). "Validating the Security and Stability of the Grader for a Programming Contest System". Olympiads in Informatics, Vol. 4, pp113-119.
- [9] Burket, Jonathan, Chapman, Peter, Becker, Tim, Ganas, Christopher, & Brumley, David (2015). "Automatic Problem Generation for Capture-the-Flag Competitions". In Proceedings of the 2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 2015).
- [10] Ihantola, Petri, Tuukka Ahoniemi, Ville Karavirta & Otto Seppälä (2010). "Review of Recent Systems for Automatic Assessment of Programming Assessments", in Proceedings of the 10th Koli calling Conference on Computing Education Research, pp.86-93.
- [11] Caiza, Julio C. & del Álamo Ramiro, José Maria (2013). "Programming Assignments Automatic Grading: Review of Tools and Implementations", pp5691-5700.
- [12] Draylson M., Souza, Felizardo, Katia R. & Barbosa, Ellen F. (2016). "A Systematic Literature Review of Assessment Tools for Programming Assignments", in Proceedings of the 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET 2016), pp. 147-156.

- [13] Douce, Christopher, Livingstone, David & Orwell James (2005). “Automatic Test-Based Assessment of Programming: A Review”, *Journal on Educational Resources in Computing (JERIC)*, Vol. 5, No. 3, Art. 4.
- [14] Keuning, Hieke, Jeuring, Johan & Heeren, Bastiaan. “Towards a systematic review of automated feedback generation for programming exercises”, in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 41-46.
- [15] Falkner, Nickolas, Vivian, Rebecca, Piper, David & Falkner, Katrina (2014). “Increasing the effectiveness of automated assessment by increasing marking granularity and feedback units”, in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, pp. 9-14.
- [16] Radošević, Danijel, Orehovački, Tihomir & Stapić Zlatko (2012). “Automatic on-line generation of student's exercises in teaching programming”, in *Proceedings of Central European Conference on Information and Intelligent Systems (CECIIS 2010)*.
- [17] Prados, Ferran, Boada, Imma, Soler, Josep & Poch, Jordi. “Automatic generation and correction of technical exercises”, in *International Conference on Engineering and Computer Education (ICECE)*, Vol. 5.

AUTHORS

Dr Sébastien Combéfis obtained his PhD in engineering in November 2013 from the Université catholique de Louvain (UCLouvain). He is currently working as a lecturer at the ECAM Brussels Engineering School, where his courses focus on computer science. He also obtained an advanced master in pedagogy in higher education in June 2014. Co-founder of the Belgian Olympiad in Informatics (be-OI) in 2010, he later introduced the Bebras contest in Belgium in 2012 and at the same time founded CSITEd. This non-profit organisation aims at promoting computer science in secondary schools.



Guillaume de Moffarts is a master student in computer science at Université catholique de Louvain (UCLouvain). He is interested in computer science and electronics, and very curious about engineering and new technologies, such as 3D printing, artificial intelligence and the internet of things. He is also involved in the CSITEd non-profit organisation, taking part on several projects it organises. He was also recently the deputy leader of a Belgian delegation to the IBU Olympiad in Informatics 2019 that was held in Skopje, North Macedonia.



SFERANET: AUTOMATIC GENERATION OF FOOTBALL HIGHLIGHTS

Vincenzo Scotti, Licia Sbattella and Roberto Tedesco

DEIB, Politecnico di Milano, Milano, Italy

ABSTRACT

We present a methodology for automatic generation of football match “highlights”, relying on the commentator voices and leveraging two multimodal NNs.

The first model (M1) classifies sequences and provides a representation of such sequences to be elaborated by the second model. M2 exploits M1 to decode unbound streams of information, generating the final set of scenes to put into the match summary.

Raw audio, along with transcriptions generated by an ASR, extracted from 369 football matches provided the source for feature extraction. We employed such features to train M1 and M2; for M1, the feature streams were split in sequences at (nearly) sentence granularity, while for M2 the entire streams were employed. The final results were promising, especially if adopted in a semi-automatic, real-world video pipeline.

KEYWORDS

Neural Networks, NLP, Voice, Text, Summarisation

1. INTRODUCTION

There are many motivations behind this project. First of all, living in a modern era where people have such easy access to information everywhere at any time has made them willing to be constantly and immediately updated. In this sense, sport fans have become more and more hungry; this can be easily seen by the amount of web sites updated with the results of each match in real time, the streaming services to watch the events, and the video sharing platforms.

However, it would require a huge amount of time to watch all the events, even for a single sport. Sport highlights, which are becoming more and more popular and heavily used by broadcasting companies, provide a recap of the most exciting parts of a sport event. It is a convenient way for knowing what happened in, for example, a round of your preferred football championship.

So far, such highlights are created by manually editing the raw video recordings, but we think there is room for improving the current video pipeline, by means of a tool that speeds up the process. In particular, we envision a semi-automatic pipeline where a tool generates a first version of the highlights, while the human editor only needs to refine them.

The aim of SFERANet (Selection of Football Events by Recorded Audio) is to train a Neural Network (NN) able to identify top moments inside a football match through the analysis of the commentators’ voices. In practice, the idea is to detect the segments where the speakers show *excitement*.

We designed two models: one able to perform sequence classification and one, encapsulating the former, able to deal with the entire event stream and giving a continuous output on the importance of the sequences of the event. Starting from that importance measure it would be possible to extract what should belong to the final event highlights.

We didn't make use of video-based features, like scoreboard graphics or sophisticated scene recognition, as the former depends on the broadcasting network and the latter requires a huge corpus.

SFERAnet is thought to be used inside a semi-automatic video pipeline, where a human editor refines the video generated by the tool.

2. RELATED WORK

Our work is based on “excitement recognition” through speech analysis, which is conceptually similar to the common task of emotion recognition. Moreover, we also leveraged literature on automatic detection of sport highlights. In the following we present some relevant papers on both topics.

2.1. Automatic Emotion Classification from Speech

Focusing our analysis to NNs, we found that the approach evolved considerably through time, especially in the last few years. End-to-end NN solutions were first brought by a work [1] proposing a simple densely-connected NN with three hidden layers to transform acoustic features –computed from utterances sub-splits– into sequence of probability distributions over the target emotion; then, probabilities were aggregated into utterance-level features using simple statistics (such as maximum, minimum, average etc.) that an Extreme Learning Machine (ELM) model used to classify the utterances.

A following work [2] proposed an improvement replacing densely-connected layers with recurrent ones; in particular, they used Long Short-Term Memory (LSTM) layers. However, they continued using local-probability aggregation into a global features vector, and Extreme Learning Machines (ELM) on top of them, to perform the classification task, as in [1].

The use of simple and naïve aggregation functions and ELMs resulted not only in a drawback for these two approaches, but also in criticism; another work [3] aimed at getting rid of the drawbacks discussed above by applying fully end-to-end pipeline without handcrafted parts in the middle. The proposed solution consisted, again, in an LSTM architecture with Connectionist Temporal Classification (CTC) approach [4] to assess the class, which proved to be useful also to deal with the different lengths of the utterances.

The last work we cite [5] used both acoustic and *linguistic* features (i.e., features coming from the textual transcription of the speech). The author compared three different models: *audio-only*, *text-only*, and *mixed*. This work, which reported an overall accuracy of 74.3% for the mixed model on the IEMOCAP corpus [6], clearly showed that a multimodal approach provides the best accuracy. For this reason, we decided to follow the same approach.

2.2. Identification of Sport Event Highlights from Speech

A first attempt proposed a system for automatic detection of baseball highlights [7], based solely on audio analysis of the commentator. The hypothesis that guided this work was that high correlation exists between speaker's voice excitement and relevant events. However, since not all events could count on the presence of speech into the background, they also considered a

baseball-specific feature: the presence of a baseball hit in the audio track. So, authors considered two distinct SVM models: identification of excited speech and identification of baseball hit candidates. Then the results from these two models were fused to provide a final estimation of the probability that the analysed segment was exciting. Eventually they reported an overall accuracy of 75%.

Another work [9] proposed an audio-based model for tennis, combining long- and short-term features. Authors presented a cascaded architecture composed of two levels. The first one, worked only on short-term features using a SVM with Radial Basis Function (RBF) kernel; on top of them a Bayesian inference model combined the results from both and generated the prediction for the considered window. The second level took both long-term features and class predictions from the first one. For both audio classifiers at the base of the model, only Mel Frequency Cepstral Coefficients (MFCC) vectors were considered as input, while the output classes were: silence, applause, and speech. Authors reported precision of 98% and recall of 96%.

In [10], instead, authors proposed a system architecture based on Piecewise Gaussian Modelling (PGM) and NNs to detect highlights, but still working only on the audio signal. In this work they tried to detach from the energy-based features, like in [7], by employing the Mel Frequency Spectral Coefficient (MFSC) representation of the audio signal as a short-term feature. The resulting feature vectors are combined through PGM to achieve a long-term description of non-overlapping, fixed-size frames of the Mel Spectrogram that are classified by the NN as “action” and “no-action” (i.e. the labels they considered for the scenes into the highlights). Authors underlined two key points about their system: it only needs a few seconds of audio samples to build the classifier, and the architecture, being based only on audio features, can be effectively employed in different sports by providing results for tennis and football. In fact, they achieved a precision of 87.2% and a recall of 97.6% in detection of highlights for tennis, and an average precision of 86.7 % in the three football matches used for tests.

2.3. Identification of Sport Event Highlights from Speech and Video

One of the first systems leveraging both audio and video clues was presented in [8], where authors proposed an audio-visual framework for sport event detection. In their work, authors pointed out some useful information, in fact they noticed how sport-specific approaches typically yielded successful results within the targeted domain because of the dramatic variances in commentary styles for different sports. However, their intention was to build a general model able to work with different sports, and this is why their data set was composed of events from football, rugby, and Gaelic football. The solution they proposed was based on a SVM classifier able to separate eventful and non-eventful sequences; for this goal the SVM took as input an aggregated features vector composed by: crowd image detection, speech band audio activity, on screen graphics tracking, motion activity measures, and field line orientation. Authors reported in the case of Gaelic football an event retrieval ratio of 97%, this was the best achieved score among their classifiers

Other relevant results in this field came from a work [11], where authors focused on visual features. The proposed system was composed of two main blocks: an unsupervised framework for event decomposition based on Hidden Markov Model (HMM), which performs diarisation of the clips (i.e. segmentation and clustering) iteratively, and a subsystem for detection of highlights, which takes out the classification task on the events to discriminate between *highlight* and *non-highlight*, based on a Linear SVM. The system worked with “easy-to-extract, low-level” visual features: the Colour Histogram (CH) and the Histogram of Oriented Gradients (HOG), which were projected to a lower dimensional space through Principal Component Analysis (PCA) in order to avoid the curse of dimensionality. The authors trained and tested the system using video clips from cricket matches (they were provided with 14000 clips that they split in half for this

purpose) and explored the results when features were considered singularly and together, achieving an equal error rate of 12,1% when using both.

More recent results [12] proposed a system for detection of rugby highlights, based on detection of acoustic events. In particular they built a multi-stage classifier, that considered two acoustic events to perform the classification task: commentator's excited speech and referee's whistle. In the proposed model a first-stage classification is applied to detect from the input audio features, then excited speech detection or whistle detection are performed; at this point, time stamps of positive classification from the second stage are stored in a buffer that is later scanned to detect if a minimum number of relevant frames are present in a fixed temporal window. Then the window is extended to cover all the relevant events for that particular scene. All the three classifiers were built using GMM, and the selected audio features were MFCC, together with their first order derivatives; in this case the reported precision was 93.4% and the recall 97.1%.

The following year, with the spread of eSports championships, a video-based highlight detector for Multiplayer Online Battle Arena (MOBA) games was proposed [13]. The author proposed various solutions for frame-wise classifiers based on CNN and RNN, considering both single and cascaded architectures, and different shapes for the output; in fact, the data set was tagged considering four different levels of highlight, starting from *non-highlight* up to *maximum relevance*. The peak performances were achieved, mostly, considering only a binary output: one of the considered games reported a precision of 83.2% and a recall of 86.3%. A point to stress out about this model is that it was designed to work with real-time video streams.

Finally, we mention H5 [14], a multimodal system for extraction of highlights from sport videos, based on sport-independent excitement measures (although in the paper only Tennis is analysed as a case study). The H5 system employed excitement markers, coming from different modalities, to score the scenes of the match; in particular, authors distinguished between audio- and text-based markers, visual markers, and game analytics. Audio-based markers were extracted through a SVM built atop deep features (coming from a Deep Convolutional Neural Network used for audio classification purposes) to classify *crowd cheer* and *commentator tone excitement*; moreover, the commentator tone was complemented by a text-based marker that matched the transcription against a dictionary of expression indicative of excitement. Visual markers, instead, were computed through two classifiers, one for *player reaction* (scenes were a player was celebrating) and the other for *facial expression* (categorized in *aggressive*, *tense*, *smiling*, and *neutral*), both obtained fine-tuning pre-trained Deep Convolutional Neural Networks for image classification. Game analytics, instead, referred to Tennis specific information; in fact, since not every point in the match has equal relevance, a side court statistician provided information distinguishing between different points (e.g. *volley winner*, *smash winner*, *match point*, etc.). The sub-models composing H5 were trained separately on manually tagged audio and video clips to extract the markers, then a separate fusion model was trained to classify the proposed clips from the markers and discriminate the highlights. To test H5 a group of users was asked to rank from 0 to 5 their interest in randomly selected clips from those proposed by H5, scores was averaged to compute the precision of the system that resulted to be 92.68%.

2.4. Comments

The results obtained by the presented works are really good; nevertheless, in many cases such systems took advantage of particular visual features, for example enabling scoreboard graphics tracking as in [8], which represented a strong aid (but made the system dependent on the broadcast network-specific graphics). In other cases, like [13], [14], that leverage DCNN to perform the analysis of the visual input, a higher computational capacity is required, not to mention the necessity of a large amount of data. In [14] authors tried to cope with this problem by fine tuning pre-trained models, but the demand of computation power remained high since the

transferred models were still huge and they still needed to build a personalized data set “by hand” to extract their markers.

Keeping on with [14], there are two other key problems to point out: they were given access to game analytics provided by side court statistician that provided information in real time about the scored points (such information is hardly available, especially considering different sports) and they had the financial capabilities to pay a group of users to score their highlights.

Moreover, as in the case of [7], metrics were computed “by hand”, in the sense that a human operator compared the resulting highlights on a *small test set* with the expected output, to circumvent errors due to misaligned highlights. Our corpus was composed of thousands of samples, making it impossible to follow the same approach. Therefore, we followed the usual cross-validation approach, without human intervention.

Another particular case is that of [9], [10], where the crowd remains silent for the whole game, except to applaud immediately after a point is scored; so, highlights were basically located by the occurrence of applause. Using such a sport-specific clue wasn’t possible in our case. Actually, in [10] an alternative for football was proposed: take advantage of crowd’s noise together with commentator’s excitement. This choice resulted, in the authors’ own words, in an approach “extremely sensitive to the spectators’ and commentators’ behaviours” for both of the analysed sports. Moreover, in our dataset the recordings of the crowd weren’t provided as a separated audio channel, and it was only possible to hear them in the background of the commentators’ voices, making it very difficult to leverage such information.

In [11], instead, two other problems were introduced. The former was that highlights were given a fixed definition (according to cricket terminology, the highlights were defined as video clips corresponding to either a *4-run*, a *6-run*, or a *wicket*) so what they actually produced was a system capable of identifying these exact events and nothing else; on the contrary, we wanted to avoid to impose a fixed rule to define the highlights. The latter was that even if the proposed system carried out event discovery within a clip, all the information from the events within that same clip were employed for classification, so the system still relied on previously cut clips of fixed length; instead, we wanted to provide a system capable of finding also the cut points of the scenes to put into the highlights.

Finally, authors of [12] employed a small data set, which was tagged manually to identify as “important” everything that they expected to trigger their system. This led, indeed, to good results, but they were a consequence of this ad-hoc choice. Differently, our corpus was based on highlights generated by professional video editors.

Summing up, the system we are presenting leverages only speech and textual features from the match commentary, which can be considered as sport-genre independent; in this way, our system can be easily ported to other sports. Moreover, as shown in previous sections, various attempts in the past years proved the presence of a relationship between the excitement in the speaker voice and the importance of the related scene; this further convinced us to follow the same approach and leverage audio features. Finally, the choice of such features results in a smaller model, easier to train and faster to run.

3. DATA SET

This section presents all the information about the data employed to train the NNs.

3.1. Provided Data

Data come from 369 football matches of the 2017-18 Italian “Serie A” championship; each video recording come with the corresponding hand-crafted highlights. Each match highlights were composed of about 20 short sequences (we call them *scenes*); see Figure 1. As the commentators’ audio tracks contained the chattering and interviews before and after the match, each video was manually searched for finding the actual starts and end of the two halves of the match.

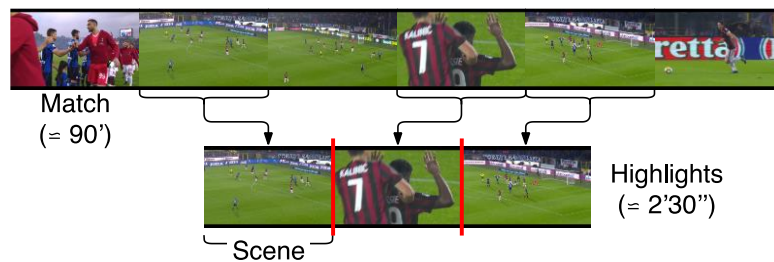


Figure 1. Match video recording and its “highlights” scenes.

3.2. Label Generation

No proper tagging of the original data was provided. To deal with this problem, we realized a tool based on perceptual hashing of images. In practice, given the video of a match and the corresponding highlights, they were both down-sampled to a grey-scale, 160×90 , 10 fps streams. Subsequently the pHash algorithm [15] was applied to each frame.

Then, the hashes of consecutive frames belonging to the same scene in the highlights were grouped together in temporal order. In this way not only the entire video scene from the highlights could be searched at once, but the results come out to be more robust since the similarity score was averaged on the entire scene. To split the highlights into scenes the similarity score between each frame and its successive was computed; in this way a drop under a fixed threshold could identify a scene change.

Each group of hashes, representing a scene, was searched computing the average similarity score against a sliding window, of the same length of the currently searched hash group, scanning the whole match. The starting frame of the window corresponding to the highest similarity score was retrieved. Once the time markers for each scene had been identified, close segments were joined; this step was necessary because sometimes either the highest similarity score didn’t lead to a perfect alignment or a part of the scene had been cut away during the editing of the summary video.

At the end, we divided and tagged the match segments:

Relevant: segments showing scenes used to compose the highlights.

Non-relevant: here are all discarded segments of the match.

3.3. Corpus Internal Structure

The corpus is composed of audio recordings of the match commentaries and, through an Automatic Speech Recognition (ASR), the corresponding transcriptions. Using the Google Speech-to-Text API permitted to obtain word-level timing alignment and speaker differentiation.

Each match contained about 2h of data but only the in-game spoken parts were considered. In this way the total amount of recordings resulted to be 640h, divided in 13h of Relevant segments and 627h of Non-relevant segments. Because of this unbalance, samples from the Non-relevant class were randomly selected in order to obtain a balanced data set so that the NN won't be badly influenced; Moreover, the "subsampling" was performed file-wise so that from each match the same amount of segments per class could be used.

In this case the term *sample* refers to the constitutive element of the data set: a scene containing audio and textual data, aligned, and coupled together. To cut the Non-relevant segments we employed a heuristic algorithm that grouped consecutive spoken parts, identified through Voice Activity Detection (VAD), in clusters of, approximatively, the same length of the Relevant ones.

These sentences were grouped into the development set, composed of segments coming from the first 50 matches, which helped to identify possible network models and hyper-parameters, and the actual data set, which used all the 369 matches segments to train, validate and test the most promising configurations and find the best one.

The content of the data sets, in terms of number of samples and duration, is reported in Table 1.

Table 1. Corpus information.

	Number of samples			Duration (sec)	
	Total	Non-relevant	Relevant	avg	std dev
All available	390932	384549	6383	3.66	2.15
Data set	12766	6383	6383	7.40	4.50
Dev set	1832	916	916	7.43	4.68

The corpus is composed by the voice of 20 different male speakers. Having a wide, different speaker presence in the data set is critically significant since it helps the network to avoid being dependent on the specific speaker's behaviour, especially for speaker-dependent features.

4. DATA PREPROCESSING AND FEATURE EXTRACTION

The raw audio signals sampled at 48 kHz were the starting point from which the input features were extracted to feed the NN-based models, this extraction process required some critical preprocessing steps that consisted, mostly, in noise suppression and downsampling; moreover in many cases, depending on the feature typology, some additional post-processing, like outlier deletion and filtering, was also required.

4.1. Audio Preprocessing

In the commentators' audio files, it was possible to hear the crowd cheering in the background. Since this noise was frequently overlapping with the voice signal to analyse, the RNNoise tool [16] was employed to get rid of it.

To reduce the amount of information to be processed, the audio tracks were down-sampled at 16kHz and the features were computed with a 20 ms wide sliding window, with a hop size of 10 ms, obtaining 100 samples per second.

Then, the preprocessing workflow executed VAD and speaker diarisation, whose results were later used for the computation of the features. In particular, results from the latter were employed to reach speaker independence. Anyhow, their use will be better explained in the following section.

4.2. Selected Features

Even though it's a common practice to leave DNNs learn the features by themselves, this approach may lead to sub-optimal solutions and incredibly complex models with subsequent waste of computational resources, as suggested by the authors of RNNoise. For these reasons, the classifiers were trained on a set of carefully selected, pre-computed features that already proved their relevance. In particular, the features we used can be classified into three groups [3]:

Prosodic. These features describe voice intonation, rhythm, and stress; we used: pitch, intensity, harmonicity, jitter, shimmer (along with their first- and second-order derivatives), chroma, silences (pauses), short-term energy with its entropy, and syllabic rhythm.

Acoustic. These features describe the spectral properties of voice; we used: MFCC, Mel bands decomposition, centroid, spread, entropy, flux, roll-off, and zero-crossing rate.

Linguistic. These features describe the semantic information contained in speech; we used word embeddings.

Prosodic and acoustic features were selected because of their correlation with perceptual aspects of the signal [17], [18], for example the pitch expresses the sentence intonation.

Apart from these features, another one represented the relative time position of the analysed segment with respect to the entire recording (the very beginning and the end of the match are very likely to be put into the highlights).

All the computed features were post-processed before being fed to the NNs, in particular we adopted a speaker-wise approach in order to obtain speaker-independent features. The post-processing steps were: standardisation, making the values of each feature in the data have zero-mean and unit-variance, outlier trimming, silent segments zeroing, and signal smoothing.

5. MODELS

Our model is actually composed of two sub-models:

- M1: for scene classification.
- M2, incorporating M1: for stream decoding, producing a continuous classification output.

The reason behind this choice stands in the structure of the former model as well as in the experiences coming from other projects. In fact, the main processing element of M1 stands in the BLSTM layer, that provides a powerful tool to analyse a scene by scanning it from start to end and vice-versa at the same time. However, as a drawback, having to deal with too long scenes, as in the case of an entire football match audio features stream, will most certainly produce poor results since the portions analysed by the forward and the backward LSTMs will be too uncorrelated.

The proposed solution for this particular problem is to train M1 separately such that it's able to *classify a single scene of known length*; after that, M2 can be trained applying transfer learning from M1, that will be used to provide a useful windowed representation. To be more detailed, the second model will perform continuous stream labelling from feature windows computed from the transferred part of the first classifier and will use a mono-directional recurrent layer to add the context from the previously analysed windows.

5.1.M1: Multimodal Scene Classifier

M1 is designed to classify a scene of variable but known length (see Figure 2); it was trained on short video scenes, namely less than a minute, however it can ideally work with unbounded ones even though performances are not ensured to be the same.

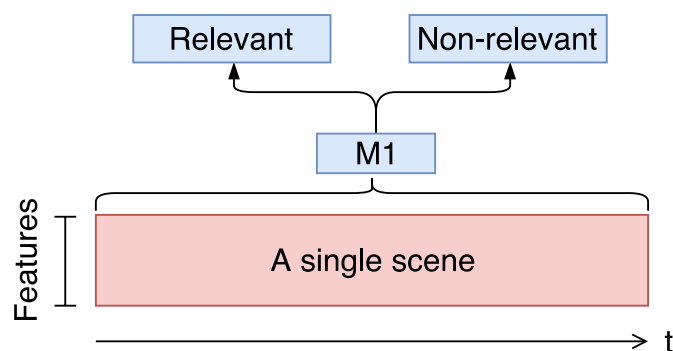


Figure 2. M1: scene classification.

As shown by Figure 3, this classifier takes the raw features of a scene and feeds them to a one-dimensional, time-distributed convolutional layer with dilation, immediately followed by a one-dimensional, time-distributed, max-pooling layer. Then, the intermediate results from the input layers are passed to a BLSTM provided with an internal attention mechanism; the BLSTM layer produces a continuous output that is weighted by the output of the attention mechanism. These weighted values are then summed up along the time to have a compressed representation of the entire scene, and the sum is scaled using a logarithmic function. The resulting intermediate representation of the entire scene is then passed to two subsequent fully-connected layers before arriving to the softmax layer with two output that represents the probabilities to belong to one of the two classes.

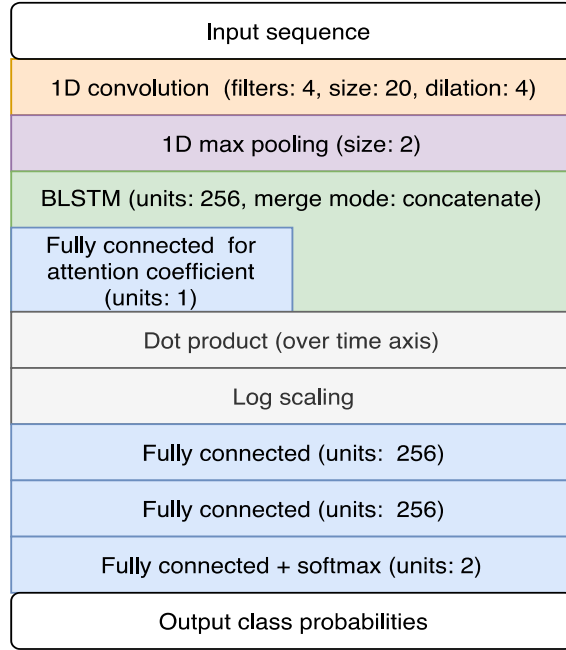


Figure 3. M1: scene classifier DNN.

5.2. M2: Multimodal Stream Decoder

M2 is designed to classify a scene of variable is designed to decode an entire stream providing a continuous classification output; it was trained on streams corresponding to entire matches.

As shown by Figure 4, this classifier takes the raw features stream as input, then slices it using a fixed-size sliding window of 7.5s with a 3.75s hop. Windows are fed in sequence to an internal time-distributed model, realised using M1, which generates an internal representation of the entire window content. These intermediate representations of the widows are then passed in sequence to an LSTM that will provide some sort of “context” among successive windows.

The continuous output of the LSTM is further elaborated by a time-distributed, fully-connected layer before the time-distributed softmax layer accomplishes the decoding task. This last layer associates the probability to belong to one of the two classes to each of the windows generated at the beginning of the pipeline. We then applied a threshold of 0.5 to identify the start and end points in time of the Relevant segments.

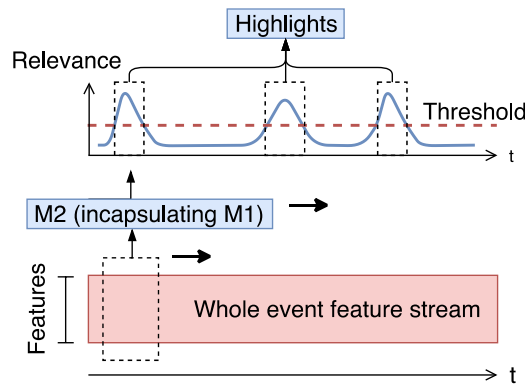


Figure 4. M2: stream decoding with sliding window.

Figure 5 shows the structure of the M2 classifier. Notice that the size of the sliding window is a parameter to be decided at “run time”, is not part of the definition of M2, and does not constraint in any way the length of the retrieved scenes.

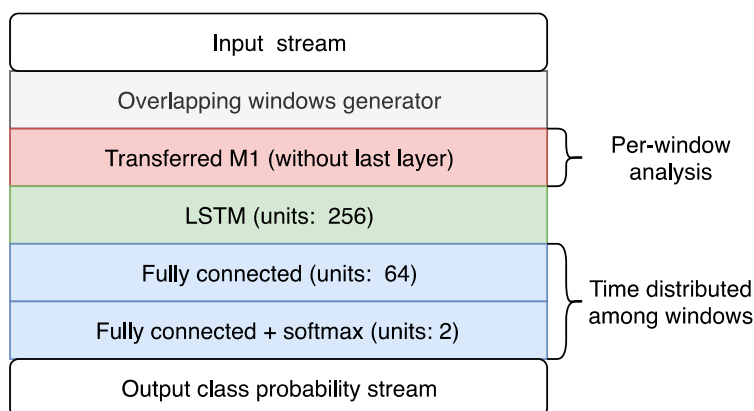


Figure 5. M2: stream decoder DNN.

As a final remark, our approach shows interesting features:

- It won't be necessary to train the entire network of M2 from scratch; in fact, thanks to transfer learning, only the top portion of the network requires training.
- Size and hop of the sliding window fed to M2 can be modified, within certain limits, without having to train the M1 network from scratch, this is due to that fact that the BLSTM layer in M1 is designed to deal with and trained on variable length scenes.

5.3. SFERAnet

Figure 6 shows SFERAnet, inside a hypothetical semi-automatic video pipeline for generation of highlights. The pre-processed speech audio is passed to an ASR and enters, along with the transcription the SFERAnet models. The result is a set of cut points (i.e., time instants where the video stream should be cut to extract the relevant scenes). Then, some video editing tool (for example, FFmpeg) could be used to generate the proposed highlights. Finally, a human expert composes the final version by means of her/his usual video editing tools.

The proposed solution for this particular problem is to train M1 separately such that it's able to classify a single scene of known length; after that, M2 can be trained applying transfer learning from M1, that will be used to provide a useful windowed representation. To be more detailed, the second model will perform *continuous stream labelling* from feature windows computed from the transferred part of the first classifier and will use a mono-directional recurrent layer to add the context from the previously analysed windows.

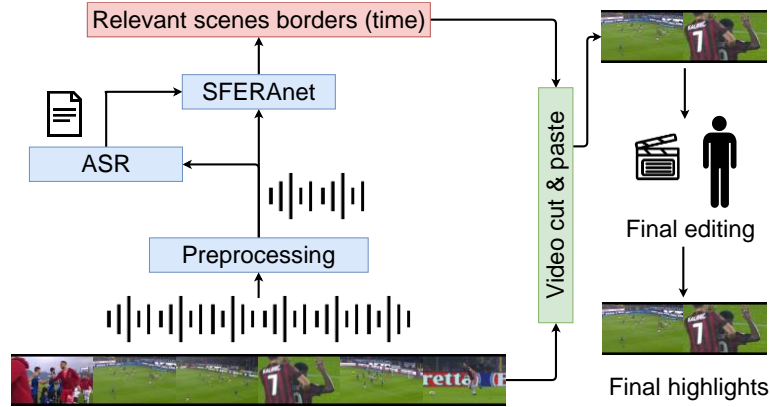


Figure 6. The SFERAnet semi-automatic pipeline.

6. TRAINING AND VALIDATION

This section will deal with the description of the training and validation process to find the best architecture.

6.1. Approach

The procedure to find the best model followed the same steps for both models. M1 was trained and validated on the samples of the data set we described in Table 1; for M2, instead, we considered as a sample the entire feature stream coming from a whole match.

The first step consisted in a grid search vowed to find the best model structure; at this stage the objective was to obtain the main structure of the model, without refining it, using a development set obtained by random-subsampling the data set.

The second step consisted in the refinement of the hyper-parameters of the best model found through the grid search, again on the development set. Differently from the previous stage, in this case there was a tree search (to lower time complexity, although at the cost of finding a sub-optimal model).

In the last step, the most promising models were compared using the results from the training on the entire data set.

In each of the presented steps, the evaluation of the model was obtained through a 10-fold cross-validation; in this way a more robust estimate of the performances could be obtained. In the train phase relative to each fold, a further split of the train set was created to be used as a validation set.

Training was performed using categorical cross-entropy as loss function, RMSProp as optimiser, and adopting the early stopping strategy. As performance metrics we computed Accuracy (using it also as a reference for early stopping), Precision, Recall, F1, Specificity, and AUC.

To deal with the class unbalance inside the data set, we considered two different approaches, depending on the model. For what concerns M1, we randomly sub-sampled the class of Non-relevant to get an equal number of scenes. For M2, instead, loss and Accuracy were weighted differently depending on the class, so that an error on the Relevant segments would be 60 times that of the Non-relevant class; the choice of that weight was done in order to reflect the available hours of recordings of each class inside the corpus.

6.2. Results

Table 2 shows quantitative values for both models. For what concerns M1, considering the best model, and in particular the results from the single best fold, it showed a high Precision. This means that M1 is particularly good in discarding the Non-relevant scenes, making it suitable for a real-world video pipeline. Moreover, it is important to stress how, with a balanced data set, Accuracy, F1 and AUC –which are used as global measure considering all the classes– show good values.

Table 2. Best results achieved by M1 and M2. The reported results are these of the models with the highest *cross-validation* (weighted) *accuracy* score.

Metric	Model					
	M1			M2		
	avg	std dev	best	avg	std dev	best
Accuracy	0.811	0.025	0.846	0.488	0.046	0.588
Weighted accuracy	-	-	-	0.682	0.020	0.715
Precision	0.884	0.035	0.900	0.032	0.003	0.038
Recall	0.719	0.063	0.778	0.852	0.033	0.822
Specificity	0.903	0.037	0.914	0.481	0.047	0.583
F1 Score	0.791	0.036	0.835	0.062	0.005	0.073
AUC	0.894	0.014	0.918	0.775	0.021	0.797

M2, instead, showed way lower scores with respect to M1. However, these scores are to be taken with a grain of salt. In fact, we found two different error categories: *model specific* and *summarisation specific*.

The model-specific errors are due to the fact that the output probability stream may be noisy around the classification threshold; in this case the problem may be fixed improving a post-processing phase. Moreover, the output probability stream may rise above the classification threshold before it is done in the target scene, and/or similarly may fall down after it, as depicted in Figure 7 (left); as scores are computed for each instance of the sliding window (i.e., every few milliseconds), even if the retrieved scene *contains* the correct one, several window instances fall outside the right interval and the computed scores are badly affected. Another typical scenario is depicted in Figure 7 (right), where a single retrieved scene contains multiple correct ones. Once again, the scores could be very low even if the model prediction is substantially correct.

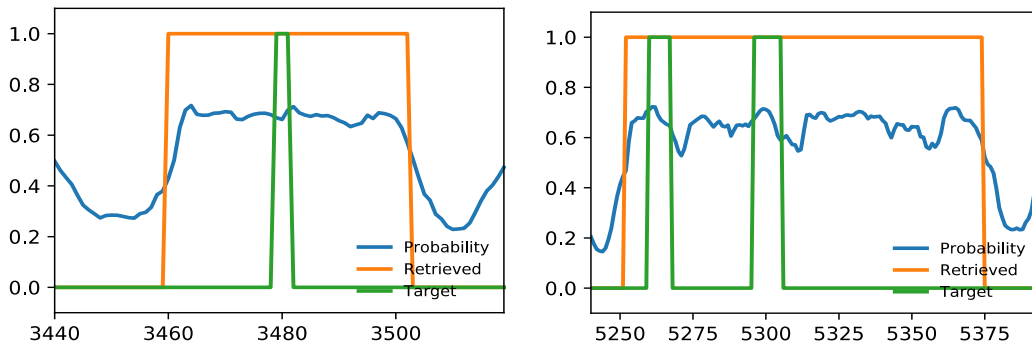


Figure 7. Output of M2 (blue), extracted scene (orange), and ground truth scene(s) (green).

The summarisation-specific errors are due to the fact that there is no “correct” metric to assess the goodness of a summary. In fact, scene cut points are somewhat arbitrary and the selection of the scenes is, to some extent, arbitrary: usually there are more relevant scenes than the ones found in the final highlight; such scenes in “excess” are cut due to time limitations (highlights shouldn’t last more than 3 minutes) but are not Non-relevant per se. For that reason, the figures we report in Table 2 are based on the usual metrics computed comparing samples in a classification task (where a sample is a decoded window of the match feature stream).

As a further validation step for M2, one should appeal to human evaluation, as some research papers we cited did. However, on one hand our corpus was too big to allow for this solution; on the other hand, a human evaluation is subjective and, in our opinion, should be avoided.

Unfortunately, in this way the problem of finding remains open but, on the other hand, it is a well-known issue even in the much more mature field of text summarisation [19]. As a final remark, better metrics could be very useful for improving the train of the model.

7. CONCLUSIONS AND FUTURE WORK

The results that are not easy to be evaluated. If M1 proved good in selecting Relevant scenes, M2 is probably not mature enough. However, in a real-world video pipeline, SFERAnet will be just a tool for a human operator. For her/him, cutting a useless scene (false positive) would be easier than add a missing scene (false negative). From this point of view, the Recall of M2 is not bad and thus SFERAnet could be actually useful, as long as it is employed in a semi-automatic pipeline.

As a future improvement, assuming to get a bigger corpus, we aim at testing more complex architectures, like GANs, which proved very powerful tools for “generation via emulation” and thus could produce more human-like highlights.

Finally, we expect to carry out some experiments on the field, by generating the highlights of matches “unseen” by SFERAnet and observing users’ reception.

REFERENCES

- [1] K. Han, D. Yu, and I. Tashev, “Speech emotion recognition using deep neural network and extreme learning machine,” in INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, Sep 2014, pp 223–227.
- [2] J. Lee and I. Tashev, “High-level feature representation using recurrent neural network for speech emotion recognition,” in INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden (Germany), Sep 2015, pp 1537–1540.
- [3] V. Chernykh and P. Prihodko, “Emotion recognition from speech with recurrent neural networks,” in *CoRR*, arXiv preprint arXiv:1701.08071, Jan 2017.
- [4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd International Conference on Machine Learning, New York (NY, USA), Jun 2006, pp. 369–376.
- [5] J. M. Origi, “PATHOSnet: parallel, audio-textual, hybrid organization for sentiment network,” Master’s thesis, Politecnico di Milano, 2018. [Online]. Available: <http://hdl.handle.net/10589/143008>

- [6] C. Busso, M. Bulut, C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: interactive emotional dyadic motion capture database," in *Language Resources and Evaluation*, vol. 42, no. 4, Nov 2008, p. 335.
- [7] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in Proceedings of the Eighth ACM International Conference on Multimedia, New York (NY, USA), Oct 2000, pp. 105–115.
- [8] D. A. Sadlier and N. E. O'Connor, "Event detection in field sports video using audio-visual features and a support vector machine," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, Oct 2005, pp. 1225–1233.
- [9] B. Zhang, W. Dou, and L. Chen, "Combining short and long term audio features for tv sports highlight detection," in *Advances in Information Retrieval*, M. Lalmas, A. MacFarlane, S. R ger, A. Tombros, T. Tsikrika, and A. Yavlinsky, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 472–475.
- [10] H. Harb, L. Chen, "Highlights detection in sports videos based on audio analysis," Oct 2009.
- [11] H. Tang, V. Kwatra, M. E. Sargin and U. Gargi, "Detecting highlights in sports videos: Cricket as a test case," in 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Jul 2011, pp. 1-6.
- [12] A. Baijal, J. Cho, W. Lee, and B.S. Ko, "Sports highlights generation based on acoustic events detection: A rugby case study," in 2015 IEEE International Conference on Consumer Electronics, Jan 2015, pp 20–23.
- [13] Y. Song, "Real-Time Video Highlights for Yahoo Esports," Nov 2016.
- [14] M. Merler, D. Joshi, K.C. Mac, Q. Nguyen, S. Hammer, J. Kent, J. Xiong, M.N. Do, J.S. Smith and R.S. Feris, "The Excitement of Sports: Automatic Highlights Using Audio/Visual Cues," in 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Jun 2018, pp. 2520-2523.
- [15] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Ismir*, vol. 32, Jan 2002, pp. 107–115.
- [16] J. M. Valin, "A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement," in *CoRR*, arXiv preprint arXiv:1709.08243, Sep 2017.
- [17] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2009.
- [18] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," in *CUIDADO IST Project Report*, vol. 54, Jan 2004, pp 1–25.
- [19] N. Schluter, "The limits of automatic summarisation according to ROUGE," in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia (Spain), Apr 2017, pp. 41–45.

AUTHORS

Vincenzo Scotti received the B.Sc. and the M.Sc. in Computer Science and Engineering from the Politecnico di Milano respectively in 2016 and 2019. He is now a PhD student in Computer Science and Engineering at the Politecnico di Milano.



Licia Sbattella Ph.D. in Computer Science, Bioengineer and Clinical Psychologist. She is Associate Professor of “Natural Language Processing” and “Personality, Team building and Leadership” at Politecnico di Milano. Since 2003 she is the Delegate of the Rector for persons with disability and psychological difficulties. She is member of the Steering Committee of UNG3ict and cooperates with the International Association of Universities (IAU) and with the Pontificia Accademia per la Vita.



Roberto Tedesco earned a M.Sc. in Computer Science, in 2001, and a Ph.D. in Computer Science, in 2006, both at Politecnico di Milano. He is contract researcher at Multi Chance Poli Team, Politecnico di Milano. His research interests are: Natural Language Processing, assistive technologies, user profiling and service customization, and e-learning.



AUTOMATION REGRESSION TESTING FOR SAS.AM WEBSITE

Harutyun Berberyan and Shahid Ali

Department of Information Technology, AGI Institute, Auckland, New Zealand

ABSTRACT

This research study is focused on a company which operated in online shopping. The company entered into the online market without proper testing. The company's site was migrated from local server to Amazon Web Services which required additional changes in its site architecture. Having automation testing especially in this case, regression test suite needs to be applied for the mentioned changes. It will be very useful for quickly testing the functionality of the site and further to validate that everything is working as expected. In order to conduct the mentioned regression testing through the test automation Selenium Webdriver was selected as a test automation tool/framework and TestNG framework was added to the test automation environment to generate comprehensive reports. After test execution the results showed that first of all the automation testing is more than 3 times faster than manual and human interaction is led to the minimum. Moreover, it proves that the core functionalities were not suffered from architectural changes although some minor bugs have been revealed during the collective execution of test cases. This research will create the regression ready solution on sas.am testers' and developers' hands also it will be a good test automation framework for all web applications created on 1C-Bitrix framework, which is getting popularity.

KEYWORDS

Amazon Web Service, Application Programming Interface, Page Object Model

1. INTRODUCTION

Nowadays number of web-based applications are deployed in different platforms and ongoing trend is to make upgrade, code modification or migration of those applications from one platform to another. In those situations, automation testing is used to perform testing by reducing the human intervention and repeatable tasks. The regression testing is one step ahead in automation testing to reveal the faults that could be increased as a result of new changes in the system.

This research study is focused on automation testing which is going to be performed on the company's website. From privacy reasons, the company name will not be disclosed. The company is located in Yerevan, Armenia and they have a supermarket chain in the same city. Today the company has a turnover of greater than \$3.5 million and 800 employees. In order to gain more profit, they operate online shopping web site "sas.am" which has entered into the market without decent testing. The website was created on 1C-Bitrix framework and supports three different languages and currencies in order to target English, Armenian and Russian speaking client's market. Also, the website is integrated with a call centre which operates 24 hours each day in order to handle clients' requests. The company sells a wide assortment of food, sweet, beverages and household products which can be ordered through the "sas.am" web site and delivered within the city by some additional cost. Brief objectives of this research are mentioned below:

- Perform website regression testing via automation testing.
- Create test automation environment utilizing Selenium WebDriver as a tool/framework for “sas.am” website.
- Integrate TestNG framework with Selenium WebDriver to simplify the “sas.am” website testing processes and generate a proper report about executed test cases.
- Create maintainable and reusable test cases using functional-decomposition approach and Page Object Modelling.
- Create a ready testing solution on testers and developers’ side to easily check the functionality as expected and fix the bugs.
- Generate and track the quality metrics for continuous improvements in product quality.
- Identify criteria for selection of functionality and write test automation script to verify and validate the following functionalities: change of site language from Armenian to English, change of currency from AMD to USD, place order of bread, rice, seafood, a search of product, sign in and sign out.

This research paper is organized as follow: Section 2 focuses on the literature review of the automation regression testing. Section 3 is focused on the research methodology for this research. Section 4 explains execution results for this research. Section 5 provides the discussion on the results of this research. In section 6 recommendations for future researches are provided. Finally, in section 7 conclusion to the research is provided.

2. LITERATURE REVIEW

The main reason for migrating “sas.am” website to AWS [1] environment was the flexibility and reliability provided by the cloud environment. The migration was supported by the fact that there are already 200 million PHP based active sites on various cloud platforms (Voda, 2014). Therefore, being PHP and MySQL based technology, the 1C-Bitrix is not an exception. However, besides the mentioned technologies 1C-Bitrix is using other components (e.g. jQuery, jQuery UI, CloudFlare) which makes architectural changes at the application level more complex. Those changes are performed in the database system, front-end layer and API (Application Programming Interface) layer (Voda, 2014). All the mentioned modifications lead to the need for regression testing which will be performed on “sas.am” website. Although lots of research are done on migrating existing PHP web sites from traditional hosting to the cloud, test automation frameworks and processes there is a lack of research about consequences related to the migration of 1C-Bitrix from VMWARE server to AWS. Also, regression testing results and practices are missing for those kinds of projects. Therefore, in order to create test automation environment and to develop regression testing for this research the literature review was conducted which is given below.

According to the research papers, where the comparison of three test automation tools was done, the most popular used tools are Selenium and QTP. However, QTP is not preferable because its license is very expensive. Although commercial versions provide full support which is not available in open-source tools, the latter has its advantage thanks to programmers who continuously add enhancements in open-source tools free of charge [9]. In addition to this research, an additional literature research has been done which proved that the most comprehensive and cost-effective open-source test automation tool is a Selenium WebDriver [7], [21].

Continuing the literature review and scaling up the research field, some literature review has been done in the past regarding test automation projects based on Selenium WebDriver. In the

mentioned research Selenium WebDriver was used in conjunction with JUnit, TestNG and POM (Page Object Model). In one research test cases are manually implemented using Java programming language and integrating Selenium WebDriver instructions with JUnit or TestNG assertions [14]. In another research is mentioned that TestNG is developed to overcome JUnit framework's some limitations. TestNG provides new features that makes it more useful than JUnit. TestNG covers all types of testing such as functional, unit and integration [7]. Despite all advantages, Selenium WebDriver does not have any built-in features to generate the test results. In order to eliminate this limitation, the TestNG framework is used with Selenium WebDriver. Eventually, in order to have more structured, optimized and reusable scripts, there is a well-known solution like Page Object Model (POM). One of POM deployment projects was done in a small Italian company (eXact learning solutions S.p.A.). The investigation has revealed the tangible benefits of applying the Page Object Model which was used in conjunction with Selenium WebDriver [14]. Although the project was done for the testing of the learning content management domain, the practice is possible to apply across many commercial projects like "sas.am".

In order to organize the mentioned testing activities, the Scrum methodology will be applied [18]. However, before adopting the mentioned methodology the following research will try to briefly cover most popular methodologies in the software development industry. Although the Waterfall Model has proven ineffective and upcoming trend in software development is the Scrum framework the Waterfall development is still widely used in software development companies [2]. Ericsson AB located in Sweden revealed the problems in the waterfall model and made the conclusion that the utilisation of waterfall model is not acceptable in large-scale projects and where the requirements are changing often. Therefore, the company changed the development model to an incremental and Agile methodology in 2005 [19]. Agile Scrum provides the speed and flexibility in product development and having this regression test research study with a short development cycle the Scrum methodology will be proposed as a solution. After summarizing these researches, it's clear that there is no evident research that highlights regression testing of the migrated 1C-Bitrix application to AWS cloud. Hence this research will be focused on regression testing of that kind of application.

3. RESEARCH METHODOLOGY

The research execution steps for this research are given below.

3.1. Functional Automation Test Plan

The functional automation test plan for "sas.am" research is shown in Table 1. The Table 1 covers the resource planning, time estimation and environment creation aspects of the research. Windows 10 was used for the installation of Google Chrome browse, Eclipse IDE, Selenium WebDriver and Java Development Kit. Those are minimal required tools for the test automation process.

Table 1. Functional automation test plan

Functional automation test plan	
Test Environment	
Operating system platform used for “sas.am” test automation	Windows 10 64 bit
Web/database server	“sas.am” is created on 1C-Bitrix framework and located in Amazon Cloud
Web browser	Google Chrome version 76.0.3
Test automation tool/framework	Selenium, version 3.14
Additional frameworks and libraries	TestNG, testng-metrics.jar
IDE	Eclipse, version 4.11
Java Development Kit	JDK version 12
Testing scope/type	Automation/regression, functional
Test resources	
Number of testers	1
Estimated start date	23.08.2019
Estimated end date	20.09.2019
Testing hours per day	6 hours
Total testing hours	150 hours

The test planning phases for “sas.am” research study is shown in Figure 1. The Figure 1 shows that the planning process starts with analysing “sas.am” website functionalities and what type of hardware and software platforms are needed for test automation. Then the suitable candidate is selected, and trainings are organized if needed. In this research, the suitable candidate is Test Automation Engineer Intern. In this phase also the test automation tool is selected for the research which is Selenium WebDriver. In schedule and estimation tester’s effort was planned for Sprint 0, Sprint 1 and Sprint 2 according to “sas.am” research. Test environment planning phase defines how the test environment is set up and who is in charge of those processes and in this case, Test Automation Engineer Intern has performed all installation and configuration tasks. The last phase describes test execution and closure of the research.

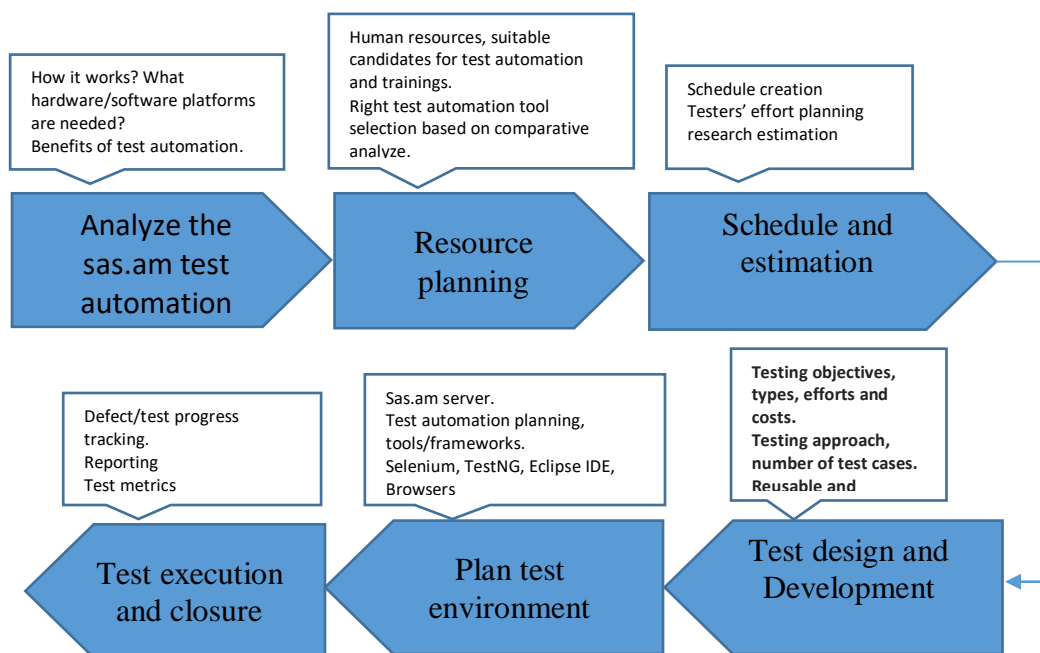


Figure 1. Planning phases

3.2. Proposed test automation framework

Proposed test automation framework for this research is shown in Figure 2, which describes the process flow and interaction between Page Factory classes, TestNG classes, Selenium WebDriver and web browsers [7]. The Page Factory class is the farther improvement to the Page Object design pattern. It is used to initialize and instantiate the elements of the Page Object. Page Factory is an inbuilt Page Object Model (POM) concept for Selenium Web Driver, but it is much optimized [14]. TestNG is integrated with Eclipse in the proposed framework in order to generate reports and to have multiple test case execution.

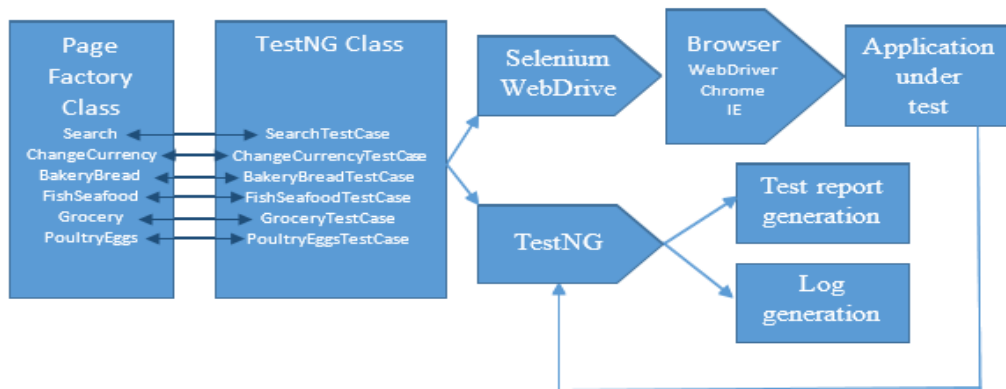


Figure 2. Proposed test automation framework

According to proposal in order to improve the maintainability and reusability of test cases, the functional-decomposition approach and Page Object Modelling (POM) is applied in this research. In the diagram the test cases are represented as Java classes. The architecture of class interaction and test execution process is shown in Figure 3.

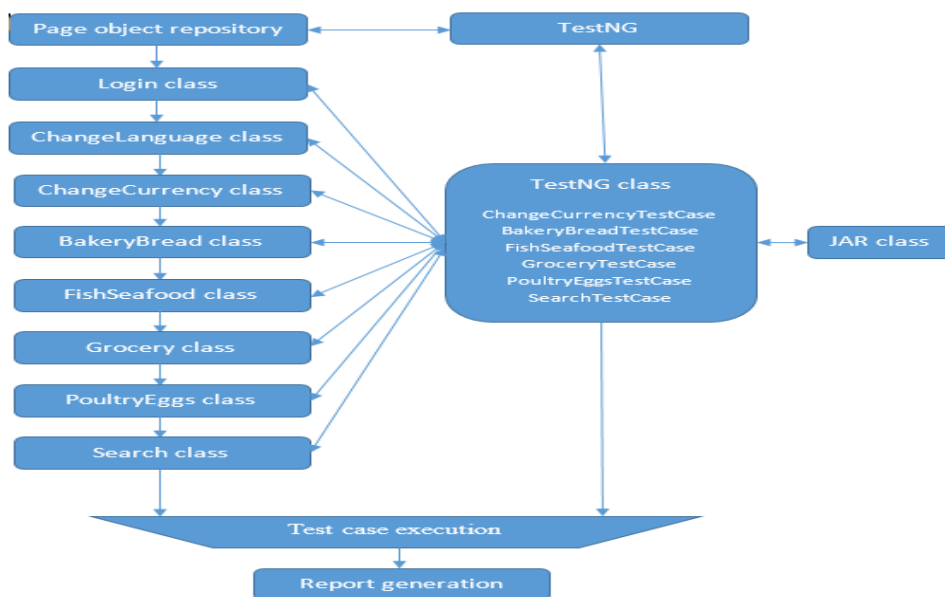


Figure 3. Architecture of class interaction, test generation and reporting

As per the POM, for every web page, the separate class has been created. Those classes have been organized in page_factory_objects package. Then another two packages have been created for the test suite and utility class which is shown in Figure 4.

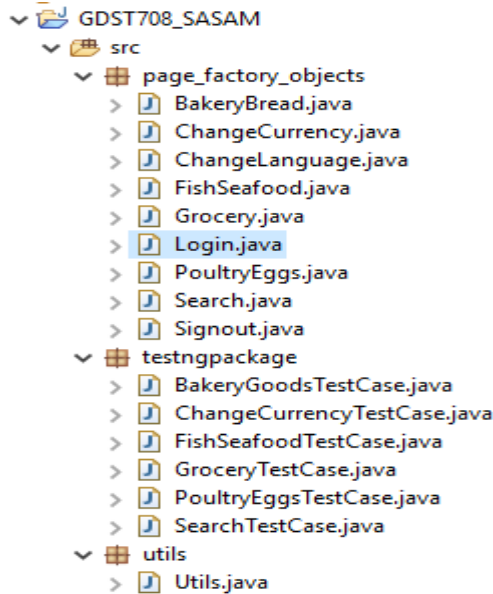


Figure 4. Organization of classes in Eclipse environment

3.3. Functional Test Cases

All selected requirements which must be done during Sprint 0, Sprint 1 and Sprint 2 are given in Table 2. Actually, the requirement is an expected behaviour of software which must be fulfilled during the testing. Thus four requirements are specified in Table 2 among those requirements are the customer registration, product order and search functionalities.

Table 2: Requirements

Req. ID	Description
Req.1	The customers shall change language and currency in “sas.am” website
Req.2	The customer shall register/sign into the “sas.am” website
Req.3	The customer shall make order of product
Req.4	The customer shall perform a search of the product

Based on functional requirements described in Table 2, the following user stories (US) are created for “sas.am” test automation research which is shown in Table 3. For example from Req.2, the following user stories have been derived:

- US3: As a customer, I want to register/login to the system so that I can add, view or change my orders
- US4: As a customer, I want to sign out from the system so that I make sure that my account is protected from other users

Table 3. User stories

User story ID	Description
US1	As a customer, I want to change the site language between three supported languages so that I can do my activities
US2	As a customer, I want to change the site currency between three supported currencies so that I can do an order of product
US3	As a customer, I want to register/login to the system so that I can add, view or change my orders
US4	As a customer, I want to sign out from the site so that I make sure that no one else can use my account
US5	As a customer, I want to make an order, view my cart or empty my cart so that I can buy an appropriate product
US6	As a customer, I want to do a search for needed product so that I can easily make an order of product

The Requirement Traceability Matrix (e.g. RTM) in Table 4 shows the mapping of user requirement with test cases. The RTM is very important because test coverage against requirements can be identified. For example, the Req.1 has been mapped to TC1 and TC2 test cases which validate the site language change functionality from Armenian to English and change currency functionality from AMD to USD. The same logic is applied on the rest requirements and test cases.

Table 4. Traceability matrix

Req. ID	Scenario ID	Test case ID	Test case description
Req.1	US1	TC1	Validate site language change functionality from Armenian to English
	US2	TC2	Validate currency change functionality from AMD to USD
Req.2	US3	TC3	Validate customer registration functionality
		TC4	Validate customer log in functionality with correct username and password
	US4	TC5	Validate customer sign out functionality
Req.3	US5	TC6	Validate order bread functionality under White Bread sub menu
		TC7	Validate order fish functionality under Fresh Fish sub menu
		TC8	Validate order rice functionality under Rice sub menu
		TC9	Validate order chicken functionality under Hens, chicken sub menu
Req.4	US6	TC10	Validate search product functionality
		TC11	Validate filter search results by min and max prices
		TC12	Validate clear search results functionality

The test case prioritization is needed because there is no lots of time and system resources to spend on full regression testing, therefore there is a need to identify which test cases should be run during Sprint 0, Sprint 1 and Sprint 2 of duration 5 weeks.

Customer registration and login functionality which are described in US3 and US4 have been given highest priority because these are the Minimum Viability Product functionalities. To avoid hips of registered user accounts into the database at present instance trying to stick with the

manual registration process instead of automation. The payment functionality has been tested manually as the credit card information can't be shared through the test automation script. The next high priority was given to order product functionality under US5 using four best-sold product statistics from the current database. Less priority was given to search product and change language functionalities under US1 and US2. Change language functionality belongs to cosmetics behaviour so it's given low priority. The user story prioritization is shown in table 5.

Table 5: User story prioritization

Task Name	User Story	Priority
Sprint 0	US3	Highest
	US4	High
Sprint 0	US5	High
Sprint 1	US5	Medium
Sprint 1	US6	Medium
	US1	Low
Sprint 2	US2	Low

3.4.Automation Test Scripts

In Test Class, the actual Selenium test automation script is written. Here also so-called page action is performed on Web Pages. For each page, its own test class is written and commented for better code readability. Test cases are written in @Test annotation which marks a method/class as a part of the test.

The small fragments of the code are used for explanation purposes. In “page object repository” nine page object classes were created, where two scripts are responsible for login/sign out functionality and seven scripts are responsible for functional test cases. As change language and login functionalities were performed during each test case, they were included in @BeforeTest annotation and the Sign out functionality was included in @AfterTest annotation. In the “testngpackage” six classes have been created where the TestNG classes (e.g. BakeryGoodsTestCase, ChangeCurrencyTestCase etc.) are controlling the test executions and the creation of an instance of Google Chrome driver, ChangeLanguage, BakeryBread, FishSeafood and for the rest classes. One of those TestNG classes is shown in Figure 5. Then TestNG Classes are controlled by testng.xml file as shown in Figure 6.


```

17 public class BakeryGoodsTestCase {
18
19     String Url = "https://www.sas.am"; //link of the sas.am web server
20     String driverPath = "C:\\Eclipse_workspace\\Webdrivers\\Chrome\\chromedriver.exe"; //the location of the web driver
21     WebDriver driver;
22
23     //declaration of the object
24     ChangeLanguage objChangeLanguage;
25     Login objLogin;
26     BakeryBread objBakeryBread;
27     Signout objSignout;
28     Utils objUtils;
29
30@
31     @BeforeSuite
32     public void setURL() {
33         System.setProperty("webdriver.chrome.driver", driverPath);
34         driver = new ChromeDriver();
35         driver.get(Url);
36         driver.manage().window().maximize();
37         System.out.println("launching chrome browser");
38     }
39@
40     @BeforeSuite
41     public void testchangelanguagethenlogin() throws InterruptedException {
42         //Create ChangeLanguage Page Object
43         objChangeLanguage = new ChangeLanguage(driver); //creating an object by calling the ChangeLanguage class's constructor.
44         objChangeLanguage.changelanguage();
45         //Create Login Page object
46         objLogin = new Login(driver); //creating an object by calling the Login class's constructor.
47         objLogin.logintosasam("harutyun19833@yahoo.com", "Abcd1234#"); //, "welcome, "

```

Figure 5. Bakery goods test case class

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
3 <suite name="Suite">
4     <test name="Test">
5         <classes>
6             <class name="testngpackage.BakeryGoodsTestCase"/>
7             <class name="testngpackage.ChangeCurrencyTestCase"/>
8             <class name="testngpackage.FishSeafoodTestCase"/>
9             <class name="testngpackage.GroceryTestCase"/>
10            <class name="testngpackage.PoultryEggsTestCase"/>
11            <class name="testngpackage.SearchTestCase"/>
12        </classes>
13    </test>
14 </suite>

```

Figure 6. TestNG xml file

In figure 7 the BakeryBread class script is depicted. The script is divided into three logical sections described below.

- Creation of the class for each functional test and finding the web elements (Figure 7)
- Performing actions on elements, like click, move to element (Figure 8)
- Creation of public method with parameters inside the class (Figure 9)

The same logic is applied on the rest of seven classes ChangeCurrency, CaseLanguage, FishSeafood, Grocery, Poultry, Search, Login and Sign out in Figure 4.

```

12 BakeryBread.java
13 public class BakeryBread {
14
15     WebDriver driver;
16
17     //"Bakery Goods" link
18     @FindBy(xpath="//*[@href='/categories/Bakery_Goods_Buns_and_Rolls_1045/']") //web element identification by @FindBy annotation
19     WebElement linkBakeryGoods;
20
21     //"White Bread" link
22     @FindBy(xpath="//*[@href='/categories/White_1064/']")
23     WebElement linkWhiteBread;
24
25     //"Bread Matnaqash" link
26     @FindBy(xpath="//*[@href='/products/Bread_Matnaqash_g_370846/']")
27     WebElement linkMatnaqash;
28
29     //"Add" link
30     @FindBy(xpath="//*[@class='btn' and @value='Add']")
31     WebElement linkAdd;
32
33     //"View full cart" link
34     @FindBy(xpath="//*[@href='/personal/cart/' and @class='light_btn']")
35     WebElement linkViewFullCart;
36
37     //"Empty cart"
38     @FindBy(xpath="//*[@class='solid_ico_clear']")
39     WebElement linkEmptyCart;
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figure 7: BakeryBread class script's part one

```

40 BakeryBread.java
41
42
43
44
45
46
47
48
49 public BakeryBread(WebDriver driver)
50 {
51     this.driver = driver; //refers to the constructor's parameter
52     PageFactory.initElements(driver, this); //static init elements of pagefactory class for initializing web element
53 }
54
55 //open "Bakery Goods, Buns and Rolls" sub menu
56 public void clkBakery(){
57     Actions actions = new Actions(driver);
58     actions.moveToElement(linkBakeryGoods);
59     actions.build().perform();
60 }
61
62 //Click on White Bread
63 public void clkWhiteBread(){
64     linkWhiteBread.click();
65 }
66
67 //Do assertion 1 "I should be presented with a screen where I can select white breads"
68 public void doAssertion(String expRes){
69     String actRes = header.getText();
70     System.out.println("Assertion White Breads: " + actRes);
71     Assert.assertEquals(actRes, expRes);
72 }
73
74 //Click on Matnaqash Bread
75 public void clkMatnaqashBread(){
76     linkMatnaqash.click();
77 }
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figure 8. BakeryBread class script's part two

```

BakeryBread.java
103
104 //Creation of public method "createorderbread" with parameters String ExpRes ....
105 public void createorderbread(String ExpRes, String ExpRes2) throws InterruptedException{
106
107 // "Bakery" link click
108 Utils.wait_four_sec();
109 this.clkBakery();
110
111 // "White bread" link click
112 Utils.wait_four_sec();
113 this.clkWhiteBread();
114
115 //Do assertion 1
116 Utils.wait_one_sec();
117 this.doAssertion(ExpRes);
118
119 // "Matnaqash bread" link click
120 Utils.wait_four_sec();
121 this.clkMatnaqashBread();
122
123 //Do assertion 2
124 Utils.wait_one_sec();
125 this.doAssertion2(ExpRes2);
126
127 // "Add item to cart" link click
128 Utils.wait_two_sec();
129 this.clkAdd();
130
131 // "View full cart" link click
132 Utils.wait_four_sec();
133 this.clkViewFullCart();
134

```

Figure 9. BakeryBread class script's part three

3.5. Executable Jar File

A Java archive (e.g. Jar) is a process of collecting all the necessary files of the "sas.am" website test research together. The main purpose of creating this file is to distribute the single executable file of sas.am test research. The script of the executable jar file is given in Figure 10. The script contains "ExecutableJarFile" public class with instance of "jarcollector" for BakeryGoodsTestCase, ChangeCurrencyTestCase etc.

```

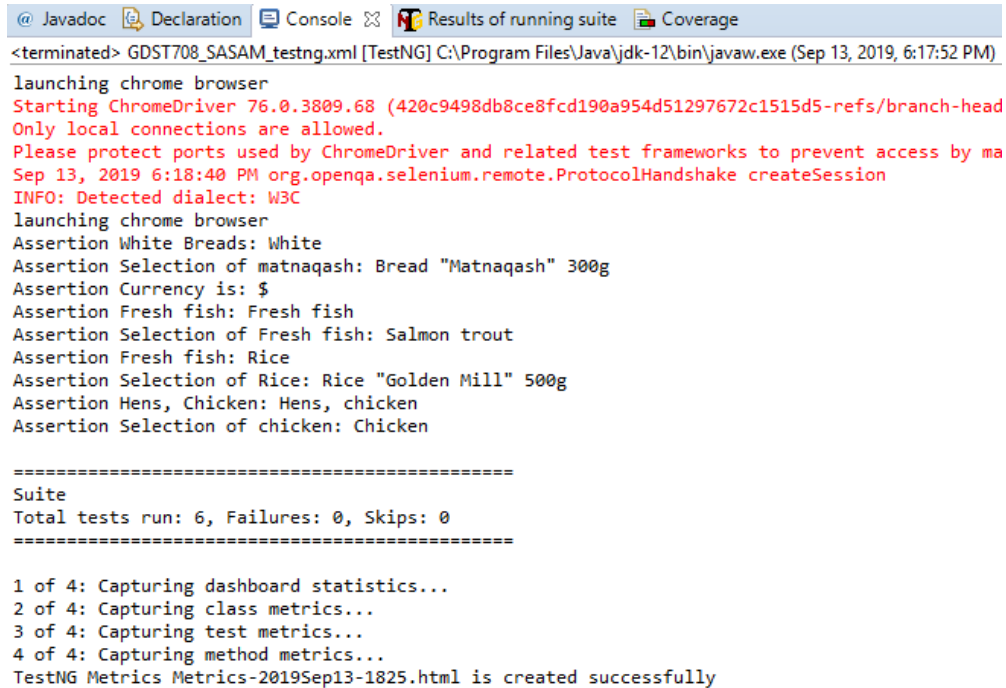
ExecutableJarFile.java
1 package testngpackage;
2
3 import org.testng.TestNG;
4
5 public class ExecutableJarFile {
6     static TestNG jarcollector;
7     public static void main(String[] args) {
8         jarcollector = new TestNG();
9         jarcollector.setTestClasses(new Class[] {
10            BakeryGoodsTestCase.class,
11            ChangeCurrencyTestCase.class,
12            FishSeafoodTestCase.class,
13            GroceryTestCase.class,
14            PoultryEggsTestCase.class,
15            SearchTestCase.class
16        });
17
18         jarcollector.run();
19     }
20
21 }

```

Figure 10. Executable jar file creation script

4. RESULTS

The Eclipse console logs are shown in Figure 11 with six passed and zero failed results. In that log, all the assertions are shown for change currency and different product order functionalities. The report also shows the file name where the generated metrics are stored and in this execution, it is Metrics-2019Sep13-1825.html file. In these logs the ChromeDriver version can be identified which is useful for debugging purposes.



```

@ Javadoc Declaration Console Results of running suite Coverage
<terminated> GDST708_SASAM_testng.xml [TestNG] C:\Program Files\Java\jdk-12\bin\javaw.exe (Sep 13, 2019, 6:17:52 PM)
launching chrome browser
Starting ChromeDriver 76.0.3809.68 (420c9498db8ce8fcd190a954d51297672c1515d5-refs/branch-head
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by ma
Sep 13, 2019 6:18:40 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
launching chrome browser
Assertion White Breads: White
Assertion Selection of matnaqash: Bread "Matnaqash" 300g
Assertion Currency is: $
Assertion Fresh fish: Fresh fish
Assertion Selection of Fresh fish: Salmon trout
Assertion Fresh fish: Rice
Assertion Selection of Rice: Rice "Golden Mill" 500g
Assertion Hens, Chicken: Hens, chicken
Assertion Selection of chicken: Chicken

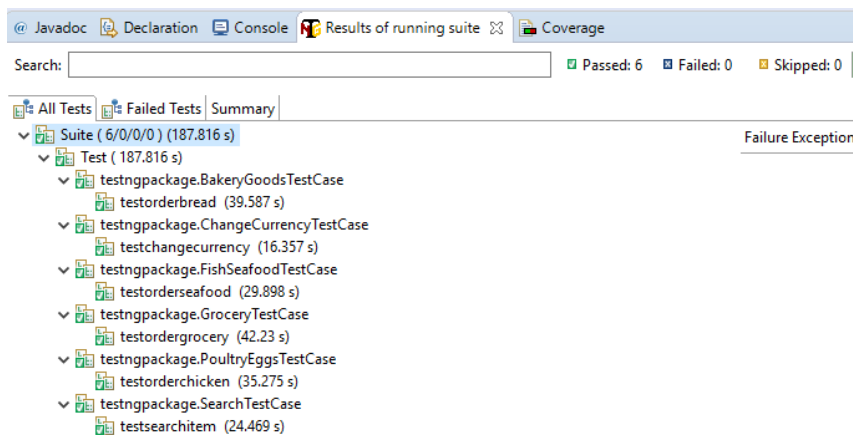
=====
Suite
Total tests run: 6, Failures: 0, Skips: 0
=====

1 of 4: Capturing dashboard statistics...
2 of 4: Capturing class metrics...
3 of 4: Capturing test metrics...
4 of 4: Capturing method metrics...
TestNG Metrics Metrics-2019Sep13-1825.html is created successfully

```

Figure 11. Console report

The precise execution time of each test case and the total execution time of the test suite are given in Figure 12. As the previous figure the Figure 12 also shows how many test cases are passed, failed or skipped. The “testngpackage” in this figure represents the package where all test classes were created.



```

@ Javadoc Declaration Console Results of running suite Coverage
Search:
Passed: 6 Failed: 0 Skipped: 0
All Tests Failed Tests Summary
Suite (6/0/0) (187.816 s) Failure Exception
  Test (187.816 s)
    testngpackage.BakeryGoodsTestCase
      testorderbread (39.587 s)
    testngpackage.ChangeCurrencyTestCase
      testchangeurrency (16.357 s)
    testngpackage.FishSeafoodTestCase
      testorderseafood (29.898 s)
    testngpackage.GroceryTestCase
      testordergrocery (42.23 s)
    testngpackage.PoultryEggsTestCase
      testorderchicken (35.275 s)
    testngpackage.SearchTestCase
      testsearchitem (24.469 s)

```

Figure 12. Results of running test suite

From the literature review we can conclude that Selenium WebDriver does not have built-in feature to generate the test results. In the framework proposed in this research TestNG is integrated with Eclipse in order to create the test report and execute test cases. This report contains all the passed and failed test cases of TestNG.

TestNG logs for timing reports of test suite are depicted in Figure 13. Actually this report has the same results as the Figure 12.

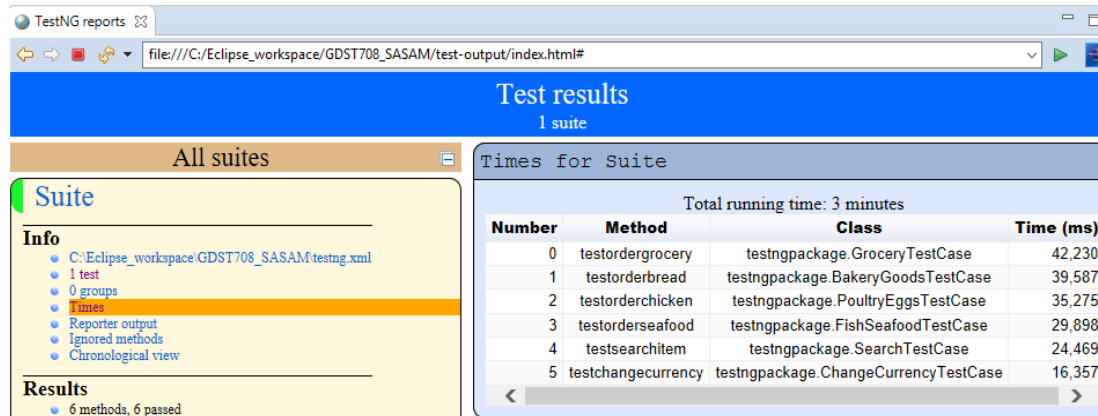


Figure 13. Timing reports for the test suite

However TestNG reports are very tedious to understand, so the “testng-metrics.jar” lib was downloaded from maven.org website and integrated into the Eclipse environment. During the execution of test cases, the “Metrics-2019Sep13-1825.html” file is generated which contains reports shown in Figure 14 to Figure 18. Figure 14 shows that six test cases have been passed and there is no failed or skipped test cases in this execution.

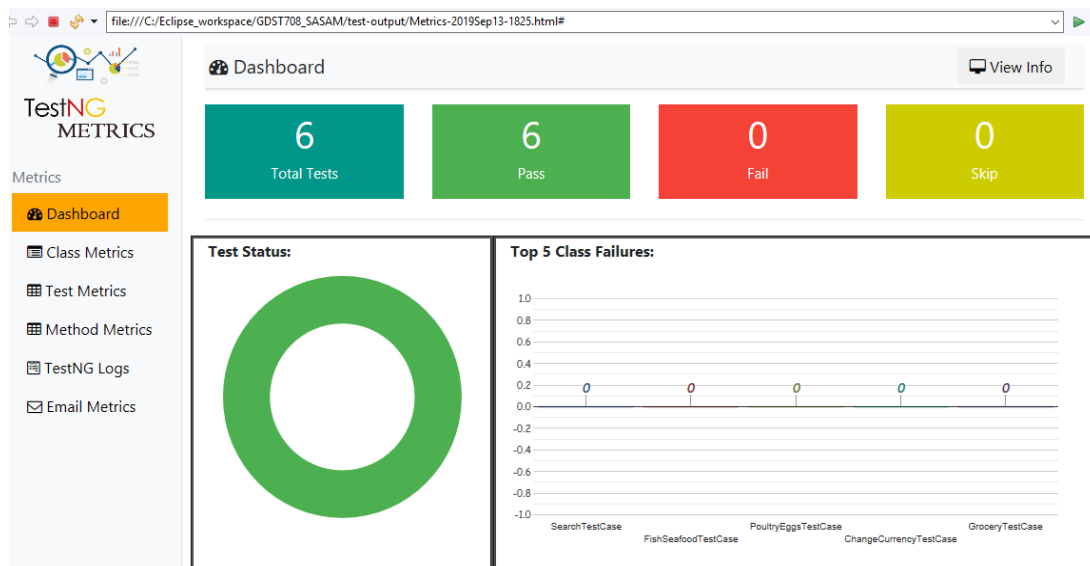


Figure 14. TestNT Dashboard report

The top ten test performances are shown in Figure 15 where the longest period of time took “testordergrocery” test execution and the shortest period of time was spent by “testchangeurrency” test execution. As mentioned, change language, login and sign out functionalities were included to all test cases except “testchangeurrency” test case, that’s why the latter is showing the smallest duration of test execution.

Top 10 Test Performance(sec):

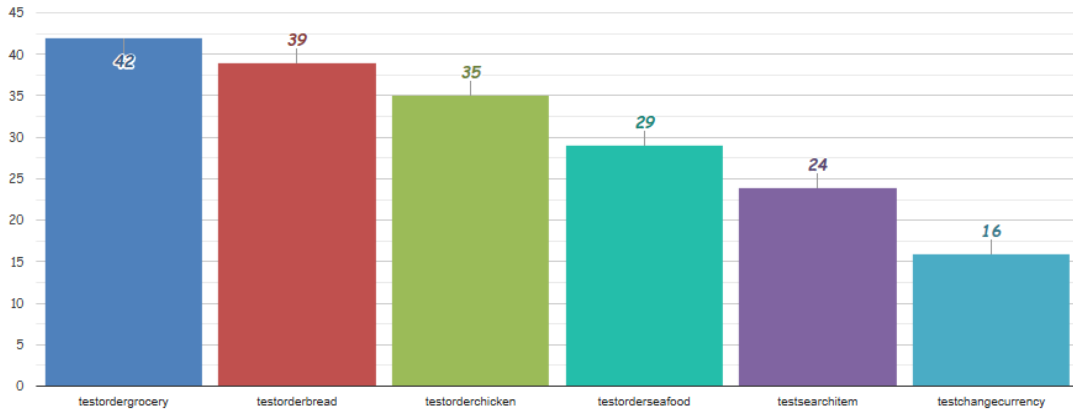


Figure 15. Top 10 Test Performances in seconds

Top 10 Config Methods Performance(sec):

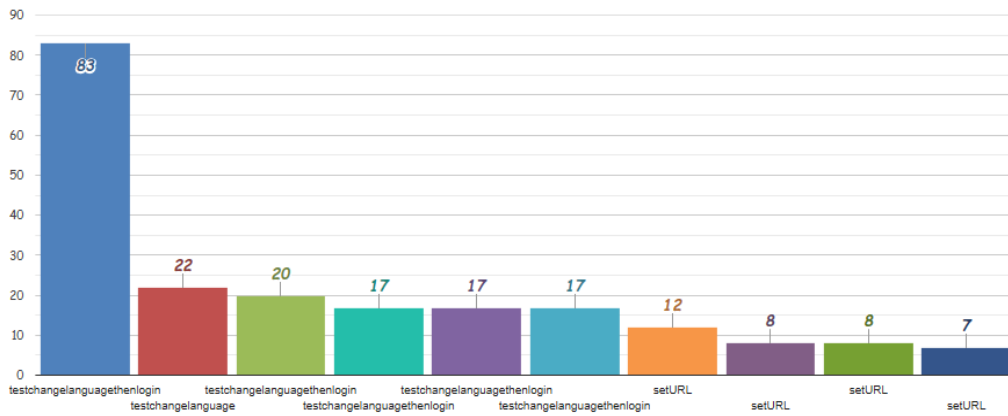


Figure 16. Top 10 Config Methods Performances in seconds

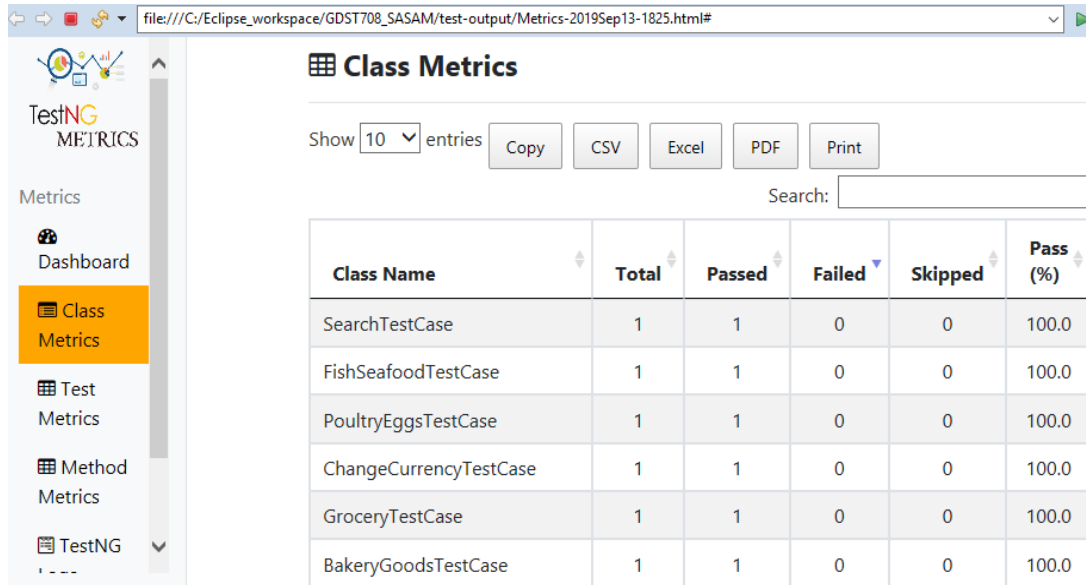


Figure 17. Class metrics

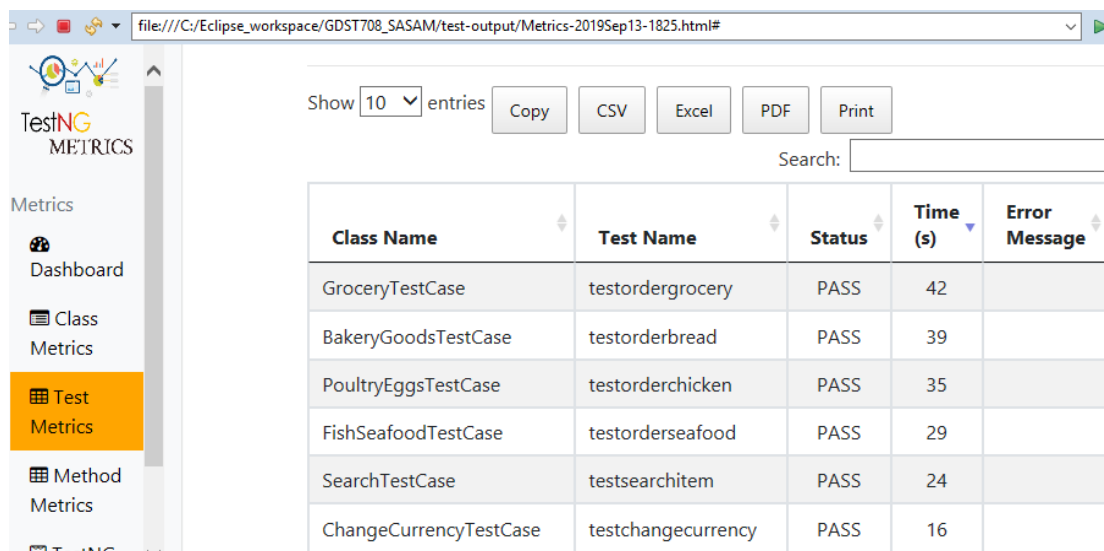


Figure 18. Test metrics

Although this report covers test automation research of “sas.am” website, the manual testing also has been performed and test execution times have been recorded for comparison purposes in Table 6. Here also change language, login and sign out functionalities have been included in test case in order to make a realistic comparison of the results between manual and automated test executions.

Table 6. Manual test execution durations of sas.am research

TestCase	Manual Testing (in seconds)
BakeryGoodsTestCase	82
ChangeCurrencyTestCase	38
FishSeafoodTestCase	63
GroceryTestCase	79
PoultryEggsTestCase	91
SearchTestCase	87
Total	440

Manual test execution screenshot is shown in Figure 19. In this screenshot, the site is opening in its default language which in this case is Armenian.

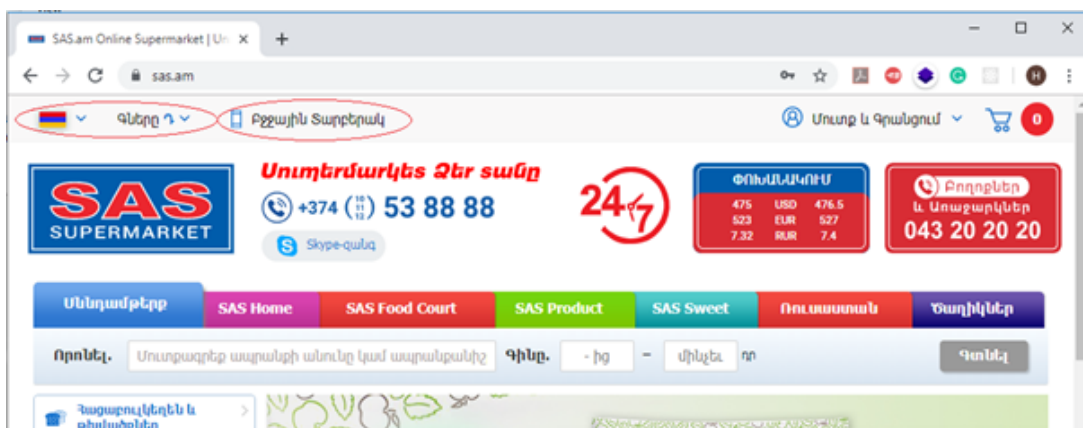


Figure 19. Screenshot taken during manual test execution

The test automation execution screenshot is shown in Figure 20, it is quite visible that Chrome web browser is being controlled by automated test software.

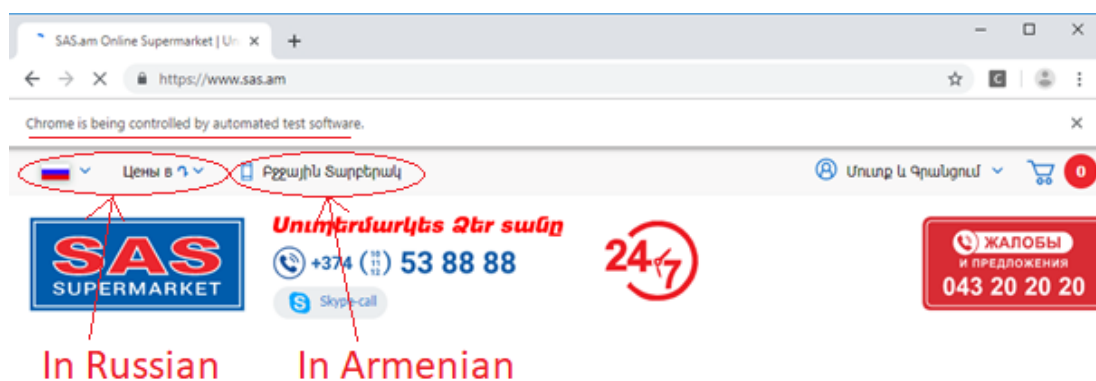


Figure 20. Screenshot taken during test automation execution

5. DISCUSSION

In this research test cases are manually implemented in Java programming language integrating Selenium WebDriver instructions with TestNG assertions.

As mentioned earlier we applied POM in this research study and Page Factory class is another form of Object Repository. Thus for each web page its own Page Object was defined. Each web element was uniquely identified and defined at the class level. Thus the “Find By” annotation was used and web element were defined so that actions were performed on them.

Web element identification has been done using custom XPath expressions. As most of sas.am site Web elements’ “id” values were unavailable or by using only one attribute like “id” it was difficult to make the element unique therefore the combination of several attributes were used. For example, the code segment in Figure 21 shows “View Full Cart” element’s identification by “href” and “class” attributes.


```
//"View full cart" link
@FindBy(xpath="//*[@href='/personal/cart/' and @class='light_btn']")
WebElement linkViewFullCart;
```

Figure 21. Identification of web element by custom XPath

For better maintainability each test case has been parameterized with different input/expected results values (e.g. String ExpRes) as shown in Figure 22 which is taken from the code fragment of “BakeryGoods” test case. This approach makes the code more optimized and reusable.

```
public void testorderbread() throws InterruptedException {
//Create "Bakery Bread" Page object
objBakeryBread = new BakeryBread(driver); //creating an object by calling the BakeryBread class's constructor.
objBakeryBread.createorderbread("white", "Bread \"Matnaqash\" 300g");
}

//Creation of public method "createorderbread" with parameters String ExpRes .....
public void createorderbread(String ExpRes, String ExpRes2) throws InterruptedException{
```



In order to interact with the login/registration form and left navigation menu of products “moveToElement” method has been used because the “click” method was useless for those case. The code fragment is shown in Figure 23. This script imitates the mouse movement towards the left vertical menu where the “Bakery Goods” link is rendered.

```
//open "Bakery Goods, Bans and Rolls" sub menu
public void clkBakery(){
Actions actions = new Actions(driver);
actions.moveToElement(linkBakeryGoods);
actions.build().perform();
}
```

Figure 22. Code fragment utilising moveToElement method

Each test case of the test suite performs various steps such as navigating web pages, ordering products, filling search forms and finally performing evaluation of a set of assertions. Hence the purpose of assertion is very critical to detect issues of the product under test. As mentioned in literature reviews TestNG provides some new functionality that makes it more powerful than JUnit. Among the advantages are assertion handling techniques (such as dependent classes, Group Test, Parameterized tests etc) which TestNG provides. The sample of assertion is shown

in Figure 24 where expected result was compared with actual result which is “White”, "Bread "Matnaqash" 300g”.

```
//Do assertion 1 "I should be presented with a screen where I can select white breads"
public void doAssertion(String expRes){
String actRes = header.getText();
System.out.println("Assertion White Breads: " + actRes);
Assert.assertEquals(actRes, expRes);
}
```

Figure 23. Assertion code sample from BakeryBread class

Utils.java file was created in order to store the Thread.sleep method, which pauses the execution for a specific period of time. For this research study four methods have been created to initiate a delay of execution one, two, three and four seconds accordingly. They also were used to initiate demonstrative delays for the tester during the execution. The script of Utils.java class is shown in Figure 25.

```
public class Utils {
    public static void wait_one_sec () throws InterruptedException {
        Thread.sleep(1000);
    }
    public static void wait_two_sec () throws InterruptedException {
        Thread.sleep(2000);
    }
    public static void wait_three_sec () throws InterruptedException {
        Thread.sleep(3000);
    }
    public static void wait_four_sec () throws InterruptedException {
        Thread.sleep(4000);
    }
}
```

Figure 24. Utils.java class file

The console report in Figure 11 shows test results for six passed test cases with their assertions. The results also show that there are no failed and skipped test cases and this means that the migration of “sas.am” web site to the Amazon Web Service didn’t affect those functionalities. These results are very useful to get a quick report about test execution.

Although Figure 14 shows that there are no failed test cases the visual observation during automated test execution revealed some minor bugs in user interface and evidence of it is given in Figure 20. In that image, the language flag and price is shown in Russian language and currency symbol and mobile version link were shown in Armenian language.

The results from Figure 18 show test metrics which contains the class names of the tests and their execution times. These timings are in direct ratio with the quantity of products contained in the corresponding page as shown in Table 7.

Table 7. Relation of test execution time to item numbers in the tested page

Test Case	Time in sec	Products (items) per page
BakeryGoodsTestCase	39	20
GroceryTestCase	42	20
PoultryEggsTestCase	35	6
FishSeafoodTestCase	29	2

Finally, the manual and automation test suites have been compared which results are shown in Table 8. During the automation testing, 8 seconds of the demonstrative delay was added to each test class which will be taken into consideration during the calculations of total test case execution duration. Thus, after doing the calculations the results showed that automation testing is 3.2 times faster than manual testing.

Table 8. Comparison of manual and automat test executions of sas.am research

TestCase	Test Automation (in seconds)	Manual Testing (in seconds)
BakeryGoodsTestCase	39	82
ChangeCurrencyTestCase	16	38
FishSeafoodTestCase	29	63
GroceryTestCase	42	79
PoultryEggsTestCase	35	91
SearchTestCase	24	87
Total	$185-6*8=137$	440

6. RECOMMENDATIONS

After conducting this research study, the recommendations are given below.

- Use Maven to easily build a project (add jars and other dependencies of the research project).
- Improve utility file by moving more methods and optimizing the existing code of sas.am test automation script.
- Execute tests parallel in AWS cloud by creating a virtual machine in the same subnetwork where the sas.am web server is located. It will help to improve the test execution performance.
- Move parametrization outside of the code and put into the CSV file. It will help to prevent direct code modification.
- Try to add more comments in the code in order to improve the readability of the script.
- Run the created regression testing on new releases of 1C-bitrix framework and compare results before making an upgrade of existing sas.am website framework.
- Put more assertions in created test scripts.

7. CONCLUSION

This report has covered test automation and regression testing framework for “sas.am” website based on Selenium WebDriver and TestNG. Although the testing and especially test automation is always recommended, the main reason for conducting regression testing was resulted because

of the recent migration of the company website to AWS platform that led to increase of the bugs in the system.

In this research study, the objectives were achieved by successfully creating an architecture of test automation framework, identification of web elements and creation of reusable automation test scripts. The research was conducted applying the Scrum methodology to expedite the testing processes. Based on prioritised user stories the test scripts were created and executed in created test automation environment. All the testing activities have been monitored and controlled by different monitoring and reporting features built into the selected test automation tool. The mentioned reporting results helped to generate and track quality metrics for continuous improvements of the product quality. Generated metrics showed testing time reduction compared with manual testing. Moreover, there were some minor bugs have been revealed by visual observation during automated test execution. The proposed framework is very significant for dynamically changing web applications like “sas.am” and it consists of reusable codes for full regression testing. Further, this research study will provide guidelines for future references regarding regression testing on migrated web applications.

REFERENCES

- [1] Amazon, E. C. (2015). Amazon web services. Available in: <http://aws.amazon.com/es/ec2/>(November 2012).
- [2] Bannink, S. (2014, January). Challenges in the Transition from Waterfall to Scrum—a Casestudy at Portbase. In *20th Twente Student Conference on Information Technology*.
- [3] Burd, B. (2017). *Java for dummies*. John Wiley & Sons.
- [4] Davies, S., & Roper, M. (2014, September). What's in a bug report?. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (p. 26). ACM.
- [5] Elallaoui, M., Nafil, K., & Touahni, R. (2016, October). Automatic generation of TestNG tests cases from UML sequence diagrams in Scrum process. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)* (pp. 65-70). IEEE.
- [6] Elbaum, S., Rothermel, G., & Penix, J. (2014, November). Techniques for improving regression testing in continuous integration development environments. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 235-245). ACM.
- [7] Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341-346.
- [8] Hamburger, V. (2016). *Building VMware Software-Defined Data Centers*. Packt Publishing Ltd.
- [9] Hanna, M., Aboutabl, A. E., & Mostafa, M. S. M. (2018). Automated Software Testing Framework for Web Applications. *International Journal of Applied Engineering Research*, 13(11), 9758-9767.
- [10] Jain, C. R., & Kaluri, R. (2015). Design of automation scripts execution application for selenium webdriver and test NG framework. *ARPJ Eng Appl Sci*, 10, 2440-2445.
- [11] Jatain, A., & Sharma, G. (2013). A systematic review of techniques for test case prioritization. *International Journal of Computer Applications*, 68(2), 38-42.
- [12] Kakaraparthi, D. (2017). Overview and Analysis of Automated Testing Tools: Ranorex, Test Complete, Selenium.
- [13] Kumar, A., & Saxena, S. (2015). Data driven testing framework using selenium WebDriver. *International Journal of Computer Applications*, 118(18).
- [14] Leotta, M., Clerissi, D., Ricca, F., & Spadaro, C. (2013, March). Improving test suites maintainability with the page object pattern: An industrial case study. In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops* (pp. 108-113). IEEE
- [15] Lewis, W. E. (2017). *Software testing and continuous quality improvement*. Auerbach publications.
- [16] Litchmore, K. A. (2016). *A comparative study of agile methods, people factors, and process factors in relation to project success* (Doctoral dissertation, Capella University).

- [17] Olsson, M. (2015). *JavaScript Quick Syntax Reference*. Apress.
- [18] Permana, P. A. G. (2015). Scrum method implementation in a software development project management. *International Journal of Advanced Computer Science and Applications*, 6(9), 198-204.
- [19] Petersen, K., Wohlin, C., & Baca, D. (2009, June). The waterfall model in large-scale development. In *International Conference on Product-Focused Software Process Improvement* (pp. 386-400). Springer, Berlin, Heidelberg.
- [20] Sharma, M., & Angmo, R. (2014). Web based automation testing and tools. *International Journal of Computer Science and Information Technologies*, 5(1), 908-912.
- [21] Sheth, T., & Singh, S. K. (2015). Software Test Automation-Approach on evaluating test automation tools. *International Journal of Scientific and Research Publications*, 5(8), 1-4.
- [22] Stikkolorum, D. R., & Chaudron, M. R. (2016, July). A Workshop for Integrating UML Modelling and Agile Development in the Classroom. In *Proceedings of the Computer Science Education Research Conference 2016* (pp. 4-11). ACM.

AUTHORS

Harutyun Berberyan was born in Yerevan, Armenia, in 1983. I received my bachelor's degree in computer systems and Informatics from the State Engineering University of Armenia, Armenia, in 2005, In the same year I joined to CISCO regional academy to get CISCO instructor courses. In 2006 I started my career as a PHP and MySQL developer in "ArdNET" company. After short period I got a job offer for IT specialist position from the biggest telecom company located in Yerevan. After getting lots of experience in the company I decided to change my job and got two job offer from pharmaceutical company KrKa d.d. and SASGROUP LLC. In KrKa I was as an IT manager and in SASGROUP LLC Network Engineer. I worked in both companies since 2018 then I moved to New Zealand to study software testing.



Dr. Shahid Ali is a senior lecturer and IT programme leader of information technology at AGI Education Ltd, Auckland, New Zealand. He has published number of research papers in ensemble learning. His expertise and research interests include machine learning, data mining ensemble learning and knowledge discovery.

TRIT: A ROBUST TRACKER BASED ON TRIPLET NETWORK

Peng Zou and Yunfei Cai

Department of Intelligent Science and Technology, Nanjing University of
Science and Technology, Nanjing, China

ABSTRACT

In this paper, a target tracking algorithm, TriT(Triplet Network Based Tracker), based on Triplet network is proposed to solve the problem of visual target tracking in complex scenes. Compared with Siamese-fc algorithm, which adopts a two-way feature extraction network, TriT uses three parallel convolutional neural networks to extract the features of the target in the first frame, the target in the previous frame and the search regions of the current frame, and then obtains the high-level semantic information of the three areas. Then, the features of the target in the first frame and the target in the previous frame are respectively convolved with the features of the current search region to obtain the similarity between each position in the search area and the target in the first frame and the target in the previous frame, so as to generate two similarity score maps. Then, interpolate and enlarge the two low-resolution score maps, and use the APCE value of the score maps as the medium to fuse the two score maps, according to which the position of the tracking target in the current frame can be located. Experiments in this paper have confirmed that, compared with some other real-time target tracking algorithms such as Siamese-fc, TriT has great advantages in tracking robustness and can effectively execute tracking tasks in complex scenes, such as illumination change, occlusion and interference of similar targets. Experimental results also show that the proposed algorithm has good real-time performance.

KEYWORDS

Target Tracking, High Robustness, Triplet Network, Score Maps Fusion

1. INTRODUCTION

With the wide application of video behavior analysis, automatic driving, human-computer interaction and other technologies, visual tracking technology has also attracted people's attention. In recent years, scholars have conducted a lot of research on it. In particular, the rise of deep learning technology has led to the development of many branches of visual tracking algorithms. However, due to the illumination change, deformation, rotation, background clutter, similar interference objects and uneven camera motion and other interference factors in the scene of visual target tracking [1,2], visual tracking is still a very challenging task in practice.

At present, the core of many tracking algorithms is to match the target image with the input frame. For an ideal tracking matching algorithm, it should provide a good match even if there are interference factors such as occlusion of the target, scale change, rotation, uneven illumination, and uneven camera movement. One solution is to explicitly model these distortions in matching by introducing affine transformation [3], probability matching [4], feature image [5], illumination invariant [6], occlusion detection [7] and other operations. However, the drawback of this method is that a modeled matching mechanism may well solve one kind of interference,

but it is likely to produce another kind of interference. In this article, we study a matching mechanism, rather than explicit modeling matches for specific interferences. We learn invariants from training videos containing various interfering factors. If training dataset is large enough, we can learn a general matching function apriori, which can deal with common interfering factors occurring in the video, such as target appearance change.

Based on the target tracking algorithm of Siamese-fc[8], this paper proposes a target tracking method named TriT based on Triplet network[9]. First, three parallel convolutional neural networks are used to extract the features of the input target in the first frame, the target in the previous frame and the search area in the current frame to obtain the high-level semantic information of the three areas. Then, the target features of the first frame and the previous frame are convolved with the features of the current search area, and the similarity between each position in the search area and the first target and the previous frame is obtained, thus generating two similarity score maps. Finally, interpolation and amplification were carried out for the two score maps with low resolution, and the APCE[10] value of the score maps was used as the medium to fuse the two score maps. Then we can get the syncretic score map according to which we can get the target position more precisely. All network models are obtained by offline pre-training, and the online tracking process does not update the model, so the frame rate can meet the requirements of real-time tracking. Experiments in this paper show that this method is more robust than the original algorithm, and its real-time performance is slightly reduced, but it can still meet the requirements of real-time tracking in most scenes.

In Section 1, we introduce the importance of target tracking technology and some basic target tracking algorithms. Then the TriT algorithm proposed in this paper is introduced. Finally, the article structure is introduced. Then in Section 2, we introduce the development of target tracking algorithm and the related work. In Section 3, we first introduce the target tracking algorithm based on Siamese network. Then the TriT target tracking algorithm, including theory and training steps and methods, is introduced in detail. Section 4 is experiment and analysis. TriT is compared with some other target tracking algorithms, and the experimental results are analyzed. Then, in Section 5, we analyze some deficiencies of TriT and propose some improvement directions.

2. RELATED WORK

Research on visual tracking algorithms has been very active in the field of computer vision in the past decades. From the initial particle filter [11] framework based algorithms to the subsequent correlation filter [12] based algorithms, the performance of tracking algorithms has been gradually improved. With the introduction of machine learning algorithms, especially deep learning algorithms, tracking algorithms have shown a trend of diversified development in recent years, and their performance and robustness have been significantly improved. The introduction of deep learning technology and the adoption of similarity measurement standard [13] can improve the accuracy and speed of the algorithm to a new level and achieve real-time and robust target tracking.

In 2016, Martin Danelljan proposed C-COT [14] algorithm. C-COT combines deepSRDCF and uses deep neural network VGGNet[15] as feature extraction network. It interpolates feature images with different resolution into continuous spatial domain by cubic interpolation, and then uses Hessian matrix [16] to obtain target positions with sub-pixel accuracy. It solves the problem of training filter in continuous space domain. The disadvantage of C-COT is the large training data and feature space, which leads to the low tracking speed. In 2017, Martin Danelljan proposed ECO[17] tracking algorithm. ECO mainly solves the problem of too large model in C-COT. It speeds up the tracking speed by reducing the correlation filtering parameters, simplifying the training set, compressing the feature space and reducing the update frequency of

the model. GOTURN[18] algorithm published by Davia Held et al. in 2015 can be regarded as the pioneer of target tracking using end-to-end deep learning model. GOTURN algorithm uses ALOV300+ video data set and detection data set in ImageNet to train a convolutional neural network based on image pair as input. The network output search area changes relative to the target location in the previous frame, so as to obtain the target location in the current frame. In 2016, Luca Bertinetto proposed a new algorithm called Siamese-fc based on deep learning tracking [8]. It uses fully convolutional Siamese network for target tracking. Its structure contains two identical fully convolutional networks, and the input is a pair of images, contain the target template and the search area. Features were extracted from the two input channels through the network, and the similarity between the template image and each position in the search area was calculated by matching the two groups of features through the template. The point with the highest similarity was the position of the target. In 2018, Anfeng He et al. from Chinese academy of sciences proposed SA-Siam[19] algorithm. It changes the network structure of Siamese-fc, adopts double Siamese network, that is, adds a Siamese network to extract the semantic features of the target object, and models the target together with the features extracted from the previous network branch, so as to improve the discrimination of the model in the target tracking task. Similar to SA-Siam, RASNet[20] proposed by Qiang Wang et al also improved the similarity measurement method based on Siamese network. RASNet uses several attention mechanisms to weight the space and channel of Siamese-fc features, and decomposes the coupling of feature extraction and discriminant analysis to improve the discriminant ability.

3. PROPOSED METHOD

3.1. Siamese Network

Standard Siamese network [21] is a kind of neural network containing two or more identical subnetwork structures. These subnetworks have the same network structure, parameters and weights. By constructing some distance measures (Euclidean distance, Manhattan distance, cosine distance), Siamese networks have become an important method in measuring learning. Hu et al. [22] applied Siamese network to face recognition and achieved 97.45% accuracy in face data set LFW.

The Siamese network is mainly composed of the following two parts (Figure 1):

- 1) Feature extraction network: Two branch networks extract features from two input values respectively. The two networks have the same structure and share weights. It is usually implemented by convolutional neural network, which includes convolutional layer, pooling layer and activation layer of some nonlinear functions.
- 2) Decision network: The role of decision network is to process the output features of the feature extraction network in the next step, so as to obtain a specific form of output. There are many kinds of decision network, which can be selected according to different task forms. For example, in some tasks, the decision network is a cascade of fully connected layers, while in others, the decision network is a series of measurement functions (euclidean distance, cosine distance, etc.) and loss functions (such as cross entropy loss, contrastive loss, etc.).

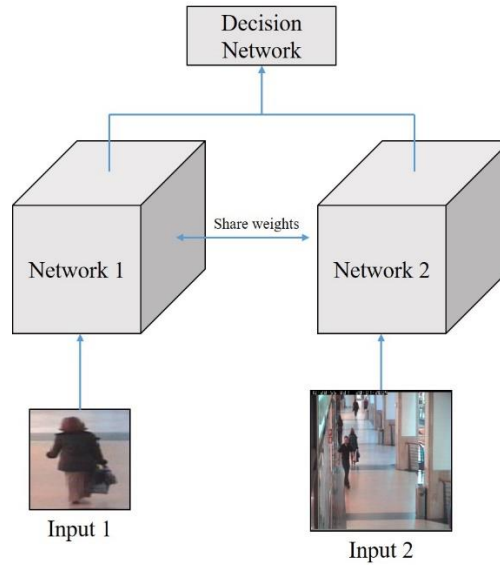


Figure1. A typical Siamese network structure

3.2. Tracking Algorithms Based On Siamese Network

The tracking algorithm based on twin neural network considers the process of tracking objects as a problem of similarity learning. It proposes to learn a mapping function $f(z, x)$. When the target image z is similar to the candidate image x , the mapping function returns a higher similarity score, otherwise it returns a lower similarity score. In order to find the new position of the target in the new frame, we need to test all possible positions and select the position in the candidate image with the greatest similarity to the tracking object as the new tracking result. Similarity mapping function $f(z, x)$ is obtained by training and learning.

In tracking phase, get a search area x centered on the center of the previous frame in the current frame, then use the feature extraction module φ (here is the convolutional neural network) to extract the convolutional features of the search area and the target area in the first frame. The mapping function $f(z, x)$ is realized through convolution operation. And then the similarity score map can be obtained, where the position corresponding to the maximum score is the new location of the target.

The specific steps could be divided into two parts:

- 1) Feature extraction of the input target in first frame and the current frame using the Siamese network, serving as φ_z and φ_x ;
- 2) Use φ_z and φ_x for feature matching, and to find the feature location with the highest feature matching score. The specific matching process is implemented with convolution operation.

In the training and tracking process, the network input is an image pair containing a large image and a small image. The small image represents the real marking box in first frame (Exemplar), while the large image represents the search area in current frame (Instance). Exemplar extraction process takes the center of the real box as the center and extracts a box of 127×127 size. When the extraction area exceeds the image, it is filled with the average RGB value of the image. Similarly, the extraction process of Instance is in the current frame. The target center of the previous frame is set as new center, and an image area of 255×255 is extracted, and the excess part is also filled with the average RGB value of the image. The output of the network is a 17×17 score map, and

each position has a score (probability value) referring to the current position as the new target center position. More accurate target center position can be obtained by adopting appropriate processing methods later.

3.3. TriT

Although the tracking algorithm based on Siamese network can well deal with some occlusion and scale changes, it is easy to fail when the background of the tracking scene is complex and there are many similar objects interfering. This kind of algorithm can distinguish the differences between different kinds of objects well, but cannot distinguish the differences between the same kinds of objects well, so it is easy to fail in tracking in some scenarios. For example, when the background is more complex or there are more similar objects interfering, such algorithm will regard the interfering object as the object to be tracked due to the lack of discrimination ability of similar objects, leading to tracking failure. Therefore, aiming at the deficiency of Siamese network, this paper proposes an improved algorithm TriT based on the network structure of Triplet network, which can simultaneously combine the first frame and the previous frame of video to comprehensively judge the current tracking results and reduce the influence of complex background and similar object interference on the tracking algorithm.

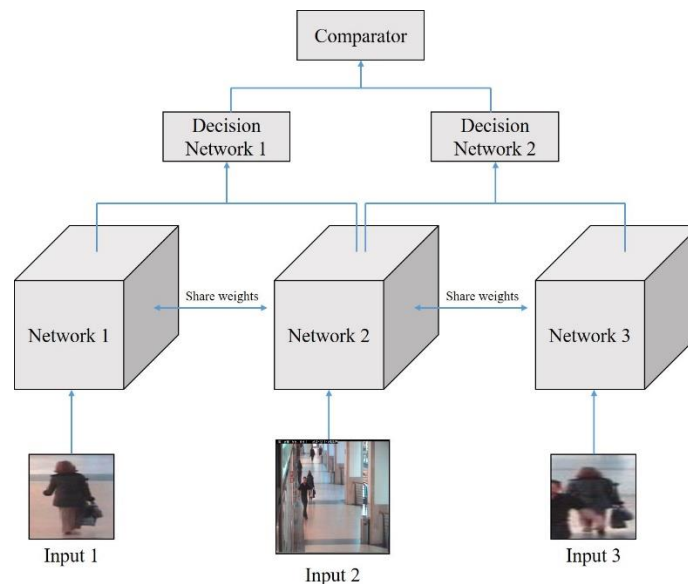


Figure 2. A Triplet network structure

Triplet network is a parallel network structure composed of three sub-neural networks. These parallel neural networks have the same network structure and the same parameters and weights. As shown in Figure 2, the Triplet network is very similar to the Siamese network, and the entire network structure can also be divided into two main parts as follows:

- 1) Feature extraction network: Three branch networks extract features from three input images respectively. The most commonly used network structure of feature extraction network is the classical convolutional neural network, such as LeNet model [23], AlexNet model [24] and VGGNet model. You can also customize some specific network structures for the feature extraction network here according to specific scenarios.
- 2) Decision network: The main function of decision network is to further process the output features of the feature extraction network to obtain a specific form of output.

In the TriT tracking model proposed in this paper, the similarity between the target in the first frame and the target in the previous frame and the search area of the current frame is calculated at the same time, and the target location is determined by merging the two score maps. The algorithm flow is shown in Figure 3.

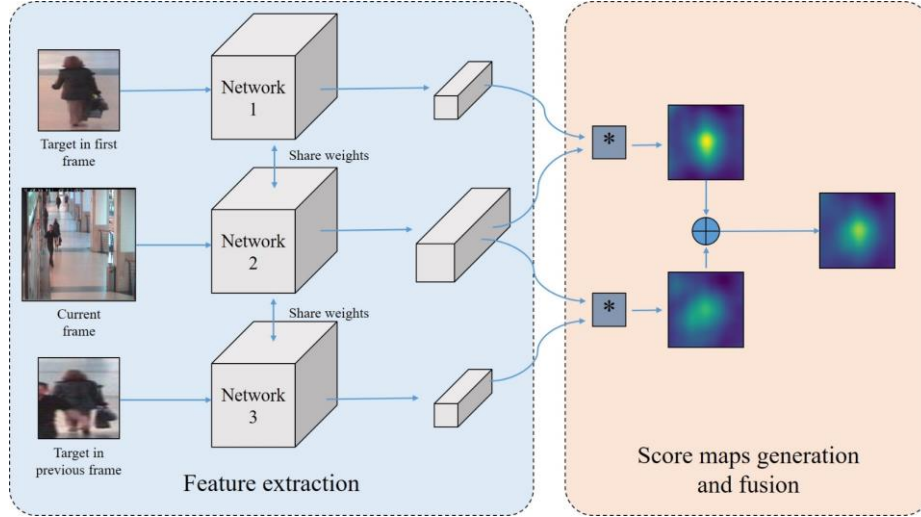


Figure 3. TriT tracking algorithm network structure

Feature Extraction. With respect to the given three inputs, namely the first frame target area z , the previous frame target area z' and the current frame search area x , we use three same fully convolutional neural networks to execute the feature extraction. Then we can get the feature output $\varphi(z)$, $\varphi(z')$ and $\varphi(x)$. The three networks appear as three parallel network structures in the model.

The advantage of fully convolutional network is that we can provide a larger search image as the input of the network, rather than the candidate images of the same size. It can calculate the similarity between all candidate sub-windows and the tracking target in one evaluation and output the result as matrix. In fact, the weights of the fully connected layer in common CNN can be remoulded into the convolutional kernel of the convolutional layer, and the fully connected layer can be transformed into the convolutional layer, so as to realize the fully convolutional network.

The feature extraction network in this paper refers to the AlexNet network structure proposed by Krizhevsky et al in [24], as shown in Fig. 4. The first convolutional layer, conv1, uses a large convolution kernel whose size is 11×11 , conv2 uses the convolution kernel whose size is 5×5 , and conv3, conv4, conv5 uses a small convolution kernel of 3×3 . The purpose of adopting such network structure is that to use a large convolution kernel in the shallow convolutional layer can quickly reduce the feature dimension and increase the receptive field, while in the deeper convolutional layer, the input feature is not too large, so the smaller convolution kernel is adopted to obtain richer semantic information. In addition, a 3×3 pooling layer is added after conv1 and conv2, with the maximum pooling and step size of 2, for further reducing the feature dimensions and maintaining the rotation invariance of the features. Except for the last layer, the network output of each layer is processed by ReLU activation function. The padding operation is not applied to the input of each layer.

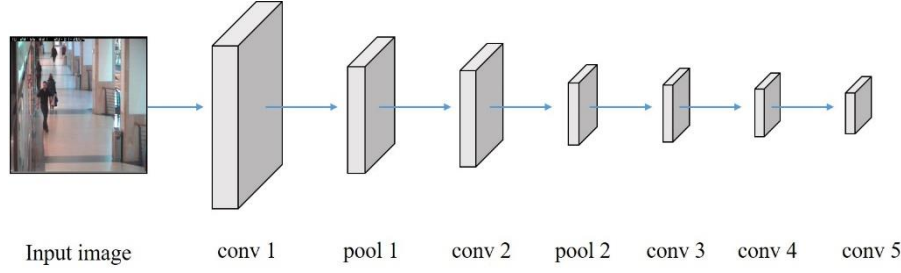


Figure 4. The feature extraction network in TriT

Training Process. The loss function uses logistic loss, in the form of:

$$\ell(y, v) = \log(1 + \exp(-yv)) \quad (1)$$

Where v represents the score of the corresponding candidate regions, $y \in \{+1, -1\}$, represents its real label. During the training, take the processing of the target in the first frame and the input image as an example. After feature extraction network, multiple candidate boxes in the input image will get multiple confidence scores, and then output them in the form of score map $v: D \rightarrow R$. Finally, the average value of logistic loss of each candidate box will be adopted as the total loss function in the form as follows:

$$L(y, v) = \frac{1}{D} \sum_{u \in D} \ell(y[u], v[u]) \quad (2)$$

Where $y[u]$ and $v[u]$ represent the real label of position u in the input image and the confidence score calculated by network.

In the training process, the stochastic gradient descent method is used to optimize the network parameters.

Score Map Generation. After feature extraction of the inputs through the fully convolutional network, the location of the target in current frame can be calculated by feature matching. We take calculating the similarity between $\varphi(z)$ and $\varphi(x)$ as an example, we multiplied the corresponding positions of the first small area of $6*6*128$ of $\varphi(x)$ and $\varphi(z)$ of size $6*6*128$ and then summed it, namely the cube convolution operation. And then we can get a similarity value, representing the similarity of the first region of $\varphi(x)$ and $\varphi(z)$. In turn, the calculation of the similarity of all $6*6*128$ in $\varphi(x)$ and $\varphi(z)$ will lead to a similarity score map $m1$. Similarly, the calculation process was used to obtain a score map $m2$ with the similarity score of $\varphi(z')$ and $\varphi(x)$. This calculation process is similar to the convolution operation of image, but it is changed from 2D to 3D. Therefore, the convolution calculation method in the convolutional neural network can be directly used for rapid implementation.

Score Map Fusion. In last section, the similarity score maps $m1$ and $m2$ are obtained by the method of convolution. Because the dimensions of extracted features are small, the score map generated is also small. This is not conducive to accurate locating. Therefore, the interpolation algorithm is firstly used to enlarge the score map to a larger dimension. In this paper, the bicubic interpolation algorithm is adopted to enlarge the score map of $17*17$ by 16 times to $272*272$, resulting in the enlarged score maps $M1$ and $M2$. Finally, the final score map M is obtained by merging the two score maps:

$$M = \lambda * M1 + (1 - \lambda) * M2 \quad (3)$$

Where λ represents the weight of the score map. The peak position of M is the target position calculated by the network.

In this paper, λ is 0.5.

4. EXPERIMENT AND ANALYSIS

In order to verify the effectiveness of the tracking algorithm TriT proposed in this paper, relevant experiments were carried out on the target tracking data set OTB100[2], and a total of 94 video sequences were tested. At the same time, the comparison experiment with the current mainstream and well-worked algorithms is carried out to draw a more convincing conclusion. Experimental environment: Ubuntu16.04 system, Intel Core i7 7800X processor (3.5ghz), 48GB of memory, NVIDIA GeForce GTX 1080Ti graphics card, TensorFlow1.4 deep learning framework, and Python programming language. In this paper, the threshold of overlap rate to judge whether it is a successful tracking is set as 0.5.

The OTB100 dataset classifies video sequences according to the challenging factors in visual target tracking, as shown in Table 1.

Table 1. Challenging factors in visual tracking.

Factor	Description
IV	Illumination Variation
SV	Scale Variation
OCC	Occlusion
DEF	Deformation
OV	Out-of-View
BC	Background Clusters
LR	Low Resolution
FM	Fast Motion
MB	Motion Blur
IPR	In-Plane-Rotation
OPR	Out-Plane-Rotation

4.1. Tracking Effect Evaluation Indicators

In this paper, two indicators are adopted to measure the experimental effect: Distance Precision (DP) and Overlap Precision (OP). DP is defined as follows: in a video sequence, the proportion of the number of frames in the video sequence whose average Euclidean distance between the tracking target location center and the real target center (marking value) is less than the set threshold. In this experiment, the threshold is set to 20 pixels. OP is defined as follows:

$$\text{score} = \frac{A_g \cap A_p}{A_g \cup A_p} \quad (4)$$

The OP reflects the overlap between the calculated location and its real location. In Equation (4), A_g represents the real position of the tracking target in the image, A_p represents the tracking target position output by the algorithm. The values of A_g and A_p are the area of the rectangular box. And the score reflects the overlap degree. The higher the value is, the higher the tracking accuracy is. Schematic diagram is shown in Figure 5.

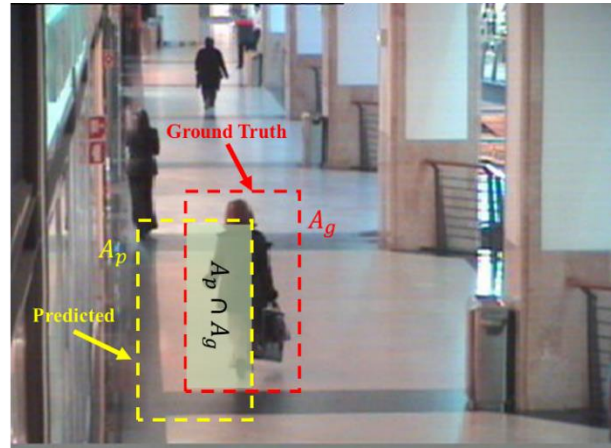


Figure 5. Schematic diagram of overlap precision

4.2. Comparison Experiment Between Trit and Siamese-Fc

The TriT tracking algorithm proposed in this paper is improved on the basis of Siamese-fc algorithm. In order to verify the effectiveness and improvement of TriT, the experimental results of comparison with the Siamese-fc algorithm are firstly analyzed.

To verify the robustness of TriT tracking algorithm, a data set containing all the challenging factors in target tracking was tested and analyzed. In Girl2, Basketball, Walking2 and Soccer video sequences, not only the appearance of the target is distorted, but also interference objects very similar to tracking targets appear in video. Figure 6 lists the comparison of TriT and Siamese-fc in Girl2, Basketball, Walking2 and Soccer video sequences respectively.

In the Girl2 video sequence, similar types of tracking target interference are generated in the image as the girl being followed passes right to left through the adult on the right. But since the little girl was not covered, both Siamese-fc and TriT were able to effectively track her. At around the 100th frame, the little girl was blocked by passers-by. After that, Siamese-fc misjudged the tracking object, but TriT was still able to effectively track the girl. In the Basketball video sequence, in 471st frame, the positions of two Basketball players with similar appearance overlap and then staggered. At this time, the Siamese-fc algorithm makes a misjudgment, treating the other player as the tracking player. By contrast, TriT does not make a misjudgment, and can still effectively track the original tracking object. In the Walking2 video sequence, a similar situation occurring in the Basketball video sequence occurs again. When a man similar to the woman tracked appears in the image, TriT can still effectively track the target, but the Siamese-fc algorithm misjudges. In Soccer video sequence, the red celebration ribbon and the scene lights have a very big interference to the tracking face. Take 292nd frame and 350th frame as examples, it can be seen that Siamese-fc algorithm wrongly locates the tracking object on another face and the cup, while TriT does not misjudge the tracking object. In addition, TriT can also be found to have a significantly higher tracking overlap accuracy than Siamese-fc in the frame where no tracking target is lost or misjudged.

Since the input of TriT algorithm contains not only the tracking information of the target in the first frame, but also the tracking information of the network output in the previous frame. On the one hand, the distortion of the target in the current frame relative to the target in the previous frame is smaller than the distortion of the target in the current frame relative to the target in the first frame. On the other hand, with the location of the target in the previous frame, the algorithm is not easy to misjudge even if there are multiple interference objects similar to the target in the

background. Therefore, TriT theoretically has more robust tracking performance than Siamese-fc, which is verified by our experiments.



Figure 6. Comparison of TriT and Siamese-fc tracking performance in Girl2, Basketball, Walking2 and Soccer video sequences in OTB100 dataset. The yellow box represents the real location of the tracking object, the blue box represents the location marked by the TriT algorithm, and the red box represents the location marked by the Siamese-fc algorithm..

4.3. Quantitative Comparisons

In order to comprehensively evaluate the performance of TriT algorithm, experiments are carried out on the OTB100 data set. With 94 video participating in the test, the DP and OP values of the algorithm are obtained and compared with the Siamese-fc algorithm and other mainstream real-time target tracking algorithms, including LCT[25], Staple[26], KCF[27] and Struck[28] algorithms. Eight representative video sequences were selected in the experiment, and the performance of each algorithm was compared. The experimental results were shown in Table 2 and Table 3.

Table 2. DP errors of TriT and other mainstream real-time tracking algorithms in the OTB100 data set (in pixels).

	TriT	Siamese-fc	LCT	Staple	KCF	Struck
Tiger2	11.3	25.3	16.7	13.7	45.1	20.3
Bird1	12.2	146.8	100.7	58.7	142.1	145.0
Soccer	14.8	47.4	62.3	68.6	39.2	81.2
Box	21.5	26.9	151.9	56.3	90.0	120.7
CarScale	16.2	5.3	53.2	33.1	88.0	101.6
Couple	15.1	5.1	19.0	28.5	44.9	23.2
Jump	48.2	57.8	165.4	189.5	129.6	135.7
Skating2-1	30.5	48.8	36.9	53.2	43.1	38.1

Table 3. OP rate of TriT and other mainstream real-time tracking algorithms in the OTB100 data set (in percent).

	TriT	Siamese-fc	LCT	Staple	KCF	Struck
Tiger2	60.4	44.6	56.9	61.4	31.5	49.0
Bird1	45.3	14.1	20.4	26.1	4.8	8.3
Soccer	54.8	21.3	12.6	20.2	39.4	17.3
Box	59.7	59.3	9.9	33.7	28.6	19.4
CarScale	76.9	77.0	67.9	76.0	41.7	41.1
Couple	59.3	68.7	41.7	51.4	19.6	50.9
Jump	28.6	20.9	4.3	4.8	8.5	9.6
Skating2-1	61.4	29.7	55.2	39.3	51.8	54.1

In Table 2 and Table 3, the sequence of CarScale, Couple and Tiger2 video sequences represents tracking in a relatively simple environment. Although the appearance of the target changes greatly, there is less interference such as objects similar to the tracked object occurring in the background. The performance of TriT algorithm is close to that of Siamese-fc algorithm, but with a slight lead in most video sequences and a performance advantage over other non-Siamese network methods in most cases. In video sequences with complex backgrounds represented by Soccer and Bird1, the tracking algorithm receives disturbances such as objects with similar appearance of tracking objects, constantly changing backgrounds, large deformation and fast change of tracking objects themselves. At this time, TriT algorithm shows great advantages over Siamese-fc and other algorithms in terms of center point error and overlap rate.

In terms of tracking speed, under the experimental conditions in this paper, the average frame rate of TriT and Siamese-fc tracking is shown in Table 4.

Table 4. Comparison of running speed between TriT and Siamese-fc (frame/second).

Tracker	TriT	Siamese-fc
Average speed	52	61

As can be seen from Table 4, TriT algorithm has a slower tracking speed, but it can still meet the requirements of real-time tracking in most scenarios. From the analysis of network structure, in the phase of feature extraction, TriT has 50% more computational load than Siamese-fc, so the tracking speed is slower than Siamese-fc.

5. CONCLUSIONS

Aiming at the problem of visual tracking in complex environment, this paper proposes a highly robust target tracking algorithm TriT. Based on the Siamese-fc algorithm, TriT adds the information of tracking target output by the previous frame to the input of the algorithm, and adopts three parallel fully convolutional neural networks for feature extraction, which is equivalent to two parallel Siamese-fc. Then the two score graphs are fused to determine the location of the tracking object in the current frame. Experiments on OTB100 data set show that TriT algorithm can still perform very robust tracking in complex environments such as illumination change, tracking object appearance change and occlusion. By contrast, Siamese-fc algorithm without the previous frame as input is very easy to misjudge the tracking object in the tracking process under a complex background. And the center point error of the target position and overlap rate calculated by TriT in the tracking process are generally better than that of Siamese-fc. TriT's tracking speed is slower than Siamese-fc due to the additional way of input, but our experiments show that TriT can still meet the requirements of real-time tracking in general tracking scenarios.

This paper mainly provides a new idea of visual tracking algorithm. Due to the simple network structure, the experimental effect is not as good as the best tracking algorithm. In later work, the method of fusing correlation filtering can be considered to update the model online, which will make the tracking algorithm more robust. Meanwhile, the feature extraction network in TriT can be improved to improve the speed of the algorithm. More reasonable loss function design can also improve the robustness of the algorithm.

REFERENCES

- [1] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2013). Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7), 1442-1468.
- [2] Wu, Y. , Lim, J. , & Yang, M. H. . (2013). Online Object Tracking: A Benchmark. *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE.
- [3] Lucas, B. D. , & Kanade, T. . (1997). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc.
- [4] Comaniciu, D. , Ramesh, V. , & Meer, P. . (2002). Real-time tracking of non-rigid objects using mean shift. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. IEEE.
- [5] Ross, D. A. , Lim, J. , Lin, R. S. , & Yang, M. H. . (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3), 125-141.

- [6] Nguyen, H. T. , & Smeulders, A. W. M. . (2006). Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision*,69(3), 277-293.
- [7] Pan, J. , & Hu, B. . (2007). Robust Occlusion Handling in Object Tracking. *IEEE Conference on Computer Vision & Pattern Recognition*. IEEE.
- [8] Bertinetto, L. , Valmadre, J. , Henriques, João F., Vedaldi, A. , & Torr, P. H. S. . (2016). Fully-convolutional siamese networks for object tracking.
- [9] Hoffer, E. , & Ailon, N. . (2014). Deep metric learning using triplet network.
- [10] Wang, M. , Liu, Y. , & Huang, Z. . (2017). Large margin object tracking with circulant feature maps.
- [11] Dai, Y. , & Liu, B. . (2015). Robust video object tracking using particle filter with likelihood based feature fusion and adaptive template updating. *Computer Science*.
- [12] Bolme, D. S. , Beveridge, J. R. , Draper, B. A. , & Lui, Y. M. . (2010). Visual object tracking using adaptive correlation filters. *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. IEEE.
- [13] Anfeng He*†, Chong Luo‡, Xinmei Tian†, & Wenjun Zeng‡. (2018). A twofold siamese network for real-time object tracking.
- [14] Danelljan, M., Robinson, A., Khan, F. S., & Felsberg, M. (2016, October). Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision* (pp. 472-488). Springer, Cham.
- [15] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [16] Mansoori, S. A. H. , Mirza, B. , & Fazel, M. . (2015). Hessian matrix, specific heats, nambu brackets, and thermodynamic geometry. *Journal of High Energy Physics*,2015(4), 115.
- [17] Danelljan, M., Bhat, G., Shahbaz Khan, F., & Felsberg, M. (2017). Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6638-6646).
- [18] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.
- [19] Anfeng He*†, Chong Luo‡, Xinmei Tian†, & Wenjun Zeng‡. (2018). A twofold siamese network for real-time object tracking.
- [20] Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., & Maybank, S. (2018). Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4854-4863).
- [21] Hong, S., You, T., Kwak, S., & Han, B. (2015, June). Online tracking by learning discriminative saliency map with convolutional neural network. In *International conference on machine learning* (pp. 597-606).
- [22] Hu, J., Lu, J., & Tan, Y. P. (2014). Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1875-1882).
- [23] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

- [24] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [25] Ma, C., Yang, X., Zhang, C., & Yang, M. H. (2015). Long-term correlation tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5388-5396).
- [26] Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., & Torr, P. H. (2016). Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1401-1409).
- [27] Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2014). High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3), 583-596.
- [28] Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., & Torr, P. H. (2015). Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10), 2096-2109.

PREDICTION AND CAUSALITY ANALYSIS OF CHURN USING DEEP LEARNING

Muzaffar Shah, Darshan Adiga, Shabir Bhat and Viveka Vyeth

Datoin Bangalore, India

ABSTRACT

In almost every type of business a retention stage is very important in the customer life cycle because according to market theory, it is always expensive to attract new customers than retaining existing ones. Thus, a churn prediction system that can predict accurately ahead of time, whether a customer will churn in the foreseeable future and also help the enterprises with the possible reasons which may cause a customer to churn is an extremely powerful tool for any marketing team. In this paper, we propose an approach to predict customer churn for non-subscription based business settings. We suggest a set of generic features that can be extracted from sales and payment data of almost all non-subscription based businesses and can be used in predicting customer churn. We have used the neural network-based Multilayer perceptron for prediction purposes. The proposed method achieves an F1-Score of 80% and a recall of 85%, comparable to the accuracy of churn prediction for subscription-based business settings. We also propose a system for causality analysis of churn, which will predict a set of causes which may have led to the customer churn and helps to derive customer retention strategies.

KEYWORDS

churn Analysis, Causality Analysis, Machine Learning, Business Analytics , Deep Neural Network

1. INTRODUCTION

Large enterprises in the competitive market mostly rely on their existing loyal customers as their large chunk of business is coming from these customers. As the market is becoming saturated day by day, the enterprises have realized that they need to focus more on retaining the existing customers [8]. While acquiring new customers is the backbone of the business growth but marketing experts suggest that equal importance should be given to retention policies[9]. As many marketing researchers suggest that retention rate can result in a significant impact on business[10]. But sometimes a customer may churn out from the enterprise which will be a loss for a company. In most of these cases, they give some prior signal before actually getting churned. Thus the main focus of the churn prediction system should be analyzing the customer behavior, such that they will identify those types of customers before they churn out and recommend some of the causes which may have led to churn. These things will help the marketing team of enterprises to adopt proper retention strategies and may help in increasing the average customer lifetime value. In turn, it will help in increasing the company's business value

in the market, as proposed by [12] that the market value of the company is a function of customer lifetime value.

First, we propose a set of generic features which can be used for all most all non-subscription business settings for developing churn prediction system. Then we present a system which will predict churn probability score for nonsubscription business settings. Finally, we introduce two methods for causality analysis to predict the possible causes that act as critical reasons for a customer to churn.

2. RELATED WORK

In recent years there has been a lot of work in this field and many systems have been proposed which will help in predicting customer churn. The advent of deep learning has also contributed to increased research in this direction. Surprisingly, there is a lack of substantial work in the direction of causality analysis of customer churn.

[4] has shown the effective use of Deep Learning to predict customer churn specifically in the Mobile Telecommunication Network. Their work concludes that a medium scale deep learning system is sufficient to predict customer churn in cellular network services, emphasizing more on unsupervised feature engineering involved in the process. Their proposed system targets generic feature vectors applicable to almost all subscription-based companies. (Using Deep Learning to Predict Customer Churn in a Mobile Telecommunication Network by Federico Castanedo) [7] are proposing a data preparation architecture that automatically comes up with more complex features and representation for the input data in the prepaid telecommunication industry. Their motivation to investigate and consider the application of deep learning as a predictive model is to avoid time-consuming feature engineering effort and ideally to increase the predictive performance of previous models. Another Deep Learning-based customer churn prediction has been proposed by [5].

There has been extensive work on conventional machine learning algorithms as well. Wei and Chiu, 2002 a proposed churn prediction system for telecommunication businesses using decision trees [3]. Burez and Van den Poel (2006) have built a churn prediction system for European pay-TV company, using the random forest algorithm [2]. Coussement and Van den Poel (2008) have applied a support vector machine for a newspaper subscription churn prediction [1].

The above systems use both conventional ML and deep learning algorithms for the churn prediction problem and most of which provide sensible results on real-world problems. The commonality among these related work is that they model churn prediction as a classification problem. So we decided to look at churn prediction as a classification problem. However, there has been very limited work on feature engineering particularly for non-subscription based businesses and none of the related work has explored the causality analysis of customer churn.

In this paper we will focus on non-subscription based business settings and will be proposing a churn prediction system, emphasizing more on the process of feature engineering. We also propose methods for causality analysis of customer churn.

3. METHODOLOGY

In the coming sections, we explain the process of feature engineering for churn prediction and we show how deep learning is used for the churn classification.

In this section, we will define important terms and definitions. Then we will give an introduction to the dataset used. After that, we will discuss the proposed feature engineering techniques, followed by a discussion on the proposed neural network used for predicting churn. In the end, we will be discussing theoretical definitions of causality analysis and how we are predicting causes of customer churn.

The churn prediction system is a data classification problem, in which we are given the historical data about customers and retention period and we have to use it to train a model which should be able to predict before a foreseeable time whether a customer will churn or not.

3.1. Type of businesses based on customer purchase behaviour

Type of business and business models has a great impact on the behavior of customers and their life cycle, which in turn affects the possibility of the customer getting churned out. Based on customer purchase behavior, there are four types of business settings:

- Contractual and discrete business:- In these types of businesses, purchase interval and sale amount both are fixed. They are also called the subscription-based business. Example:- Postpaid telecommunication, Netflix
- Contractual and continuous business:- Where purchase interval is fixed but the sale amount can vary. Example:- Credit card businesses
- Non-contractual and discrete business:- When the purchase interval can vary but the sale amount is fixed. Example:- Oil and Gas retail businesses
- Non-contractual and continuous business:- Where both sale amount and purchase interval can vary. Example:- Grocery stores, Retailer businesses

Further, each of these business categories can be classified as business-to-business or business-to-consumer. However, the methods proposed in this work apply to both of these sub-categories of businesses. Most of the churn prediction systems until now have been proposed for Contractual and discrete and Contractual and continuous types of businesses. In this paper, we will be focusing on business type Non-contractual and continuous business but our system can be generalized to other types of businesses as well.

3.2. Payment and sales data

Different enterprises have different representations and structures of sales and payment data in their system. Also, the set of fields in the data representing the sales and payment will vary from business to business. Despite the diversity in the schema and representations of the data, at the least, every enterprise will have data about sale and payment of a dealer, per day, per month or on

some other time granularity. The proposed method makes use of this minimal set of data to generate enough features to build a basic churn prediction system.

As mentioned earlier, churn prediction is a typical classification problem in which we generate training data from historical sales and payment data, one sample per customer and assign a label to each customer whether he has churned or not. We have used sales and payment data of a business-to-business large-scale non-contractual and continuous business in all of our experiments.

The consumer of this business is called the dealer who acts as a tradesperson between the business and the end-user. In our experiments, we intend to predict the churn of these dealers based on their sales and payment data.

4. FEATURE ENGINEERING

It is often said that "data is the fuel of machine learning". This is not quite true: data is like the crude oil of machine learning which means it has to be refined into features or predictor variables, to be useful for training a model. Without relevant features, you can not train an accurate model, no matter how complex the machine learning algorithm. Algorithms are pretty naive by themselves and cannot work out of the box on raw data. Hence the need for engineering meaningful features from raw data is of utmost importance which can be understood and consumed by these algorithms.

The process of extracting features from a raw dataset is called feature engineering. Though one of the main concerns of deep neural networks is to automate the process of feature engineering (Philip Spanoudes, Thomson Nguyen [5]), i.e to shift the burden of preparing and processing of feature vectors to the underlying learning system itself. But this does not mean that the data preprocessing, feature extraction, and feature engineering are irrelevant when one uses deep learning, particularly when we are dealing with direct numeric data like sales, finance, sensory data, etc. We cannot just feed any type of sales data to the neural network and expect it to generate the churn prediction problem.

In our work, we propose a set of generic features that can be extracted from sales and payment data of almost all non-subscription based businesses. The proposed feature set in our method is based on a well-known marketing technique called RFM analysis. RFM analysis, which is an abbreviation for Recency, Frequency, Monetary, is a marketing technique used to determine quantitatively which customers are the more valuable ones, by examining how recently a customer has purchased (recency), how often they purchase (frequency), and how much the customer spends (monetary) [5]. RFM analysis is based on the marketing axiom that "80% of your business comes from 20% of your customers".

We derive a set of features that represent the RMF properties of sales and payment data. Since almost all the businesses capture the sales and payment data to capture the RMF information, our feature engineering steps can be reproduced very easily. We use below set of fields that can be seen in sales and payment datasets, for the feature generation

Table 1. Common fields used from Sales data.

Field	Description
Sales date	The date on which sale has happened
Customer id	Unique business id of the dealer
Amount	Sale on that date

We use monthly granularity in our experiments because the monthly sale and payment of the business data we are using are sufficient for our system to generate features. Using the above common fields from Sales and Payment data we derived below features per dealer.

Table 2. Common fields used from Payment data.

Field	Description
Payment date	The date on which payment has happened
Customer id	Unique business id of the dealer
Amount	Sale on that date

Table 3. Features used from sales data

S. No.	Feature Name	Description	Importance
1	Overall sales of dealer	The overall lifetime sale of the dealer	Indicates the importance of a dealer based on his contribution to revenue. This represents the monetary part of RMF
2	The monthly average sale of dealer	The lifetime monthly average sales of the dealer	Represents the monetary part of RMP
3	Overall sales growth	Sales growth of dealer	Defines whether the business with the dealer is increasing or decreasing. This represents the monetary part of RMF
4	Average sale gap in days	Sales gap indicates how often the dealer is buying from the enterprise	Represents the frequency factor of RMF
5	Overall sales frequency growth	Defines whether the sales gap is increasing or decreasing and the quantity of changed sales frequency	Represents the frequency part of RMF
6	The recent average sale of dealer	Average sales of the dealer in the recent period	Represents recency part of RMF
7	Relative recent sales	The ratio of recent average sale and overall average sale	Represents recency part of RMF
8	Relative recent sale gap	The ratio of recent average sale gap and overall sales gap	Represents recency part of RMF
9	Recent average sale gap in days	Average sales gap of the dealer in the recent period	Indicates the recency part of RMF

10	Recent sale growth of dealer	Sales growth in the recent period	Indicates the recency part of RMF
11	Recent sales frequency growth	This defines whether the sales gap is increasing or decreasing in the recent period	Indicates the recency part of RMF

Table 4. Features used from payment data

S. No.	Feature Name	Description	Importance
1	Overall payment of the dealer	The lifetime payment of the dealer	Indicates the importance of the dealer based on revenue and it also represents the monetary part of RMF
2	The average payment per month of dealer	The lifetime monthly average payment of the dealer	Represents the monetary part of RMP
3	Average payment gap in days.	Payment gap indicates how often the dealer is paying	Indicates the frequency part of RMF
4	Overall payment gap growth.	Defines whether the payment gap is increasing or decreasing and the quantity of changed payment	Represents the frequency factor of RMF
5	Recent Payment of dealer	Payment received from the dealer in the recent period	Indicates the recency part of RMF
6	Relative recent average payment	The ratio of recent average payment and overall average payment	Represents recency part of RMF
7	Recent average payment gap in days	Payment gap in the recent period	Indicates the recency part of RMF
8	Relative recent average payment gap	The ratio of recent average payment gap and overall payment gap	Represents recency part of RMF

We calculate the growths by approximating a line on monthly data (sales, sales gap, payment, etc) of the dealer and then calculating the slope of that line. For example, if the slope of the line on the monthly sales data is positive then sales growth is positive and vice-versa. For creating training data, we consider the dealers who have not done any sales transactions from the last 120 days as churned dealers. The logic is also described in [11]. This number of days can be customized according to different use cases and particularly depends upon the average sales gap of the customer in that business.

The target field in our experiments indicates whether a customer has churned or not. To overcome the unavailability of this data we have chosen a novel solution. The data we have used contains sales and payment transactions from April 2016 to December 2018. The dealers who have not transacted from 30th September are considered as churned. One important point to note here is that we should not have to take the last transaction date of the dealer as a training feature, because we are deriving our target field from that date and also, this will not be available during inference. Also, in all the above-mentioned features, recency is considered as 6 months, it can also vary from problem to problem.

4.1. Churn prediction using Feed-forward neural network

Most of the related work has achieved remarkable accuracy in churn prediction [5, 6] using neural networks. We used 5 layers (MLP) feed-forward back-propagation neural network, with the input layer containing all of the input fields or features used to predict the outcome variable. The output layer contains an output field which is the target of the prediction. We have used the TensorFlow library in our neural network experiments. The architecture of the neural network used for churn prediction is mentioned in Table 5.

Table 5. Network architecture.

Hidden layers	200
Activation function	Relu in hidden layers and softmax in the output layer
Learning rate	0.01
Optimization Algorithm	Sigmoid Cross entropy

Exploring the most suitable algorithms or more complex networks for churn prediction is out the scope of our work as we are emphasizing more on feature engineering and causality analysis of churn.

5. EXPERIMENTS AND RESULTS

We have used sales data of the enterprise which has around 6000 customers, leading to 6000 training examples. In the data set, 80% of customers were still active and around 20% were already churned as per 120 days condition already mentioned in section 3.3. We have divided the data into three sets by random sampling: 5000 samples in the training set, 500 samples for evaluation set and the remaining 500 for the test set. Evaluation set is used to tune the network and algorithm to maximize the accuracy and other metrics while a test set is used to get the accuracy of the system on unseen data. Precision for churn class indicates the confidence of the model in predicting the churn. The recall represents the percentage of predicted churned customers out of the total churned customers. The F1-score is the weighted sum of the precision and recall providing a balanced score to evaluate the churn model.

We trained the network for 500 epochs and after a certain number of experiments we came up with the best configuration as mentioned in the earlier section, which resulted in an accuracy of 79.65% on the test set. Evaluation metrics of the best configuration are given in Table 6.

Table 6. Evaluation of the Churn analysis model

	Non-churn (%)	Churn class(%)
Precision	77.47	71.18
Recall	85.29	89.52
F1 Score	80.63	63.38

Generally, the number of churn events will be far less than the number of non-churn events. A representative of such a population, our training data also has fewer churn samples than the non-churn samples, making the data biased. Thus, the results are favorable towards non-churn class than the churn class. Apart from just classifying a dealer as churned or non-churned, the output of the model represents the probability of churning. This will be useful for prioritizing the dealers based on their probability of churning out, for further marketing analysis.

As per our literature study, there has been no work related to churn prediction for non-subscription based businesses and there is no benchmark dataset to compare our results with.

6. CAUSALITY ANALYSIS AND PREDICTING CAUSE OF CHURN

6.1. Introduction

Causality analysis can be defined as a process of predicting the root cause of an event. Using causality analysis we can predict the probability of a factor is a cause of certain events. Causal analysis is different from the normal regression analysis. In regression analysis, the goal is to develop a model for making predictions about the dependent variable, based on the observed values of the independent variables whereas in causal analysis, the independent variables are regarded as causes of the dependent variable.

The causal study aims to determine whether a particular independent variable affects the dependent variable and to estimate the magnitude of that effect if any. The independent variable which affects the dependent variable or may play a role in the occurrence of an event can be called a causal variable. If the variables simultaneously increase or decrease with the dependent variable but it does not cause the occurrence of the dependent variable, it may be called just a symptom of event.

In churn prediction, the causal variables are those types of features that cause customers to churn. While these causality features will vary from business to business, for the particular business setting considered in this work, the following factors can be considered as major causal factors:

1. Number of Complaints
2. Salesman or point of contact changed
3. Orders canceled due to understock
4. Returns due to defective material

6.2. How to identify the possible causes

During the data collection and preparation phase, the business representative, who has good knowledge of the business, is asked to identify the factors which may lead to churn. For instance, in this particular example, the factors identified are the ones mentioned above. Now the system

will predict the contributing score for each factor for a given customer who has a high churn risk score as per the churn prediction algorithm. The data set used in this section for experimentation is the same as used in the churn prediction system above.

6.3. Methods for predicting causality

Literature suggests different methods for generic causality analysis such as Bayesian Causal networks based model [24] [25], Structural Equation Models (SEM) [13] [14][21] [22] [23], Counterfactuals based model [13] [14] [15]. In this work, we will be proposing two models specifically for causality analysis of customer churn: Counterfactuals based causal model and predicting cause by using Bayes theorem.

6.3.1. Counterfactuals based causal model

In this model, we backtrack the event and try to undo the effects of a possible cause and then re-run the experiment [16] [17]. The goal is to find if undoing the effect will change the magnitude of the event or will not stop the event from happening [18] [19] [20]. In our example the model will work as:

- Take all the customers for whom our churn prediction system have proposed a high churn probability
- Now for every customer take all the possible causal factors and try to undo their effect one by one in the data (data used for generating features for churn)
- Now after the change in data, regenerate the features for that customer and re-calculate the churn probability of that customer
- If the recalculated churn probability is more or equal to actual probability, then we can say that this is not the major cause of churn
- But if this is less than previous churn probability, then we can say that this is one of the causes of churn magnitude of the cause can be calculated from the difference of churn probability

Example of Counterfactual method Let's consider a bunch of customers who have high churn probability as predicted by our churn prediction system. Now let us calculate the effect of the causal factor 'orders getting blocked' on the possible churning of these customers, so as a first step we have to try to undo the effect of blocked orders. Thus we will assume that the orders were not blocked, so we will add the blocked order amount to his sales data. With the change in sales data, payment data is also going to get affected, so we will try to add the same amount periodically in payment data as well. Thus the data features (used for predicting churn) have changed. So, we will apply our churn prediction on the manipulated data (feature sets). If the probability of churn gets increased or remains unchanged we can say the factor 'orders blocked' was not the causal factor, but if the probability decreases we can claim that the blocking of orders was one of the causes for customer to get churned and the magnitude of the probability of the cause can be calculated by difference of churn probabilities. One important thing to mention here is that the possibility of going through and undoing the changes is not possible in every situation with high accuracy. So, for the cases where we found it difficult to undo the effect of the cause,

like the 'change in point of contact or salesperson', some other causal model like Bayes theorem may come to the rescue.

6.3.2. Using Bayes theorem to find the probability of cause

Let us assume that y is an event that a particular cause has led to the customer churn and let x represent that customer c has been churned with churn score P_c . Thus $p(y | x=P_c)$ represents that a particular factor being the possible cause of churn for the customer c with churn score P_c . So, the problem remains to calculate $p(y | x=P_c)$. According to the Bayes theorem [26], the posterior probability of an event given the predictor can be calculated as:

$$p(y|x) = (p(x|y) * p(y)) / p(x)$$

Where

- $p(y)$ is the prior probability of an event.
- $p(x | y)$ is the likelihood which is the probability of predictor given event.
- $p(x)$ is the prior probability of predictor.
- We can further apply a Gaussian distribution [27] calculation to calculate the probability distribution of a continuous variable as shown below.

$$PDF(x, mean, sd) = (1 / (\sqrt{2 * PI} * sd)) * \exp(-((x-mean)^2)/(2*sd^2))$$

Example of the Bayes method In this example, let us consider one of the above-mentioned causes like "blocked orders due to insufficient stock" and let us assume that the blocked orders due to insufficient stock will be considered as the cause, if the block per month is 25% of order for a dealer for more than 6 months. Let y be the event that blocked orders is the cause of churn (with above-mentioned conditions) for a customer c and x be predictor indicating that the customer c has been churned by churn probability P_c . Therefore $p(y | x=P_c)$ can be defined as the probability of blocking of the orders being a cause of churn for a given customer whose churn probability score is P_c .

To calculate the above probability by Bayes theorem, we need to calculate $p(x=P_c | y)$ from the data, which means we have to calculate the probability of a customer having a churn score of P_c given his orders have been blocked (with the above condition). Since P_c is a continuous value we need to calculate the probability distribution of using the Gaussian distribution formula. From the data, we have to calculate the mean and standard deviation of churn score having 25% of blocked orders for more than 6 months.

Given a churn score P_c of a customer c , we will use that in equation (2) with mean and standard deviation calculated from the data and we will get $p(x=P_c | y)$. From the data, we also have to calculate $p(y)$, which indicates the probability that a customer's order is getting blocked. We also have to calculate $p(x)$ which is the posterior probability of the predictor. The $p(x)$ indicates the probability of customers having a given churn score P_c . Since this is also a continuous variable, so we will use the Gaussian distribution formula to calculate the value. For this, we have to calculate the mean and standard deviation of the churn score of customers in data. Given the

churn score P_c of the customer, we can calculate the $p(x)$. Using the above-calculated values in equation (2), we can get the probability of event say 'block being a cause' given a churn score P_c . If the probability is greater than 0.5 we can say it is a possible cause.

Unlike the counterfactual method, the Bayes method can be applied to almost all the scenarios. Bayes method of finding the cause is more applicable when we are interested in the magnitude of change in probability score.

6.4. Result of Causality analysis

We have experimented with both the proposed methods using our data on two different causes of customer churn. For order blockage (order blockage being the cause) we have used counterfactuals and found that if we unblock these orders (as proposed above) it will change the payment and sales data of these customers which in turn changes the features used for churn prediction.

When we used these new feature vectors to predict the churn, around 45% of customers, whose orders were blocked and had predicted as churned before unblocking, have been predicted as non-churned after unblocking. We also found that there has been an average reduction of 0.3 in the overall churn score of customers using counterfactuals on blocked orders.

For the cause of 'returns due to defect', we used the Bayes method as proposed above and found that about 64% of customers who returned their orders due to defect and have churned have 'returns due to defect' as a possible cause of churning. The causality analysis of our dataset shows the below results.

Table 7. Results of causality

Cause	Method	Average Change in Churn Score	Percentage of Customers churned due to the cause
Order Blockage	Counterfactual	0.3	45%
Order returned due to defect	Bayes Theorem	-	64%

The above results are specific to a particular dataset and business and can not be generalized. Also, the magnitude of the numbers does not depict the betterness of method or algorithm, instead they act as strong significance of various causes leading to churn in that particular dataset. As per our knowledge, the causality analysis of customer churn has not been explored until now and there is a lack of baseline experiments to compare against.

7. CONCLUSION

Churn prediction and customer retention is an important problem in the subscription as well as non-subscription based businesses and can have a considerable impact on the business if ignored. In this paper, we have investigated and developed a generic representation of feature engineering steps that could be applied to any non-subscription based business for predicting churn. We have also proposed an MLP network that is used to predict the churn probability of customers using the

data prepared though the above feature engineering process. Furthermore, based on the prediction results, it has been proved that we have achieved considerable good results targeted specifically for non-subscription based business.

The novel feature engineering process proposed by this paper has shown significant results in the churn prediction. In the second part of this paper, we have proposed two different methods to predict the possible causes of customer churn which would help the businesses to improve their customer relationship by knowing the possible causes of customer churn.

Extending the feature engineering techniques described and the use of business-specific datasets other than just payment and sales data to improve results could be a possible future work. Other causality analysis techniques, particularly for churn analysis, need to be explored further. The proposed system of causality analysis can be applied to different domains like machine failure detection, classification and diagnosis of patients, etc and there is a huge scope for experimenting with different causality analysis algorithms like Bayes Casual Network, SCM, etc.

REFERENCES

- [1] Coussement, K., Van den Poel, D. (2008) Churn prediction in subscription services: An application of support vector machines while comparing two parameter selection techniques, *Expert Systems with Applications*, 34, 313-327[1]
- [2] Burez, J., Van den Poel, D. (2006) CRM at a Pay-TV Company: Using analytical models to reduce customer attrition by targeted marketing for subscription services, *Expert Systems with Applications*, 32(2), 277-288.
- [3] Wei, C. P., Chiu, I. T. (2002). Turning telecommunications call details to churn prediction: A data mining approach, *Expert Systems with Applications*, 23, 103-112.
- [4] Anuj Sharma and Dr Prabin Kumar A Neural network-based Approach for Predicting Customer Churn in Cellular Network Service, *International Journal of Computer Applications* (0975 - 8887)
- [5] APhilip Spanoudes and Thomson Nguyen Deep Learning in Customer Churn Prediction: Unsupervised Feature Learning on Abstract Company Independent Feature Vectors by Philip Spanoudes and Thomson Nguyen , <https://arxiv.org/pdf/1703.03869.pdf>
- [6] Anuj Sharma and Dr Prabin Kumar A Neural network-based Approach for Predicting Customer Churn in Cellular Network Service, *International Journal of Computer Applications* (0975 - 8887)
- [7] AYen-Liang, Mi-Hao, Shin-Yi and Kwei Tang Discovering recency, frequency, and monetary (RFM) sequential patterns from customers purchasing data , *Electronic Commerce Research and Applications journal* (<https://fardapaper.ir/mohavaha/uploads/2017/12/FardapaperDiscovering-recency-frequency-and-monetary-RFM.pdf>)
- [8] AYen-Liang, Mi-Hao, Shin-Yi and Kwei Tang Using Deep learning to predict customer churn in Mobile telecommunication network , <http://wiseathena.com/>
- [9] Hadden, J., Tiwari, A., Roy, R., Ruta, D. (2005) German mobile cellular telecommunications market. *Telecommunications Policy*, 25, 249-269. , Computer assisted customer churn management: State-of-the-art and future trends. *Computers Operations Research*, 34, 2902- 2917.
- [10] Mozer, M., Wolniewicz, R., Grimes, D., Johnson, E., Kaushansky, H. (2000) Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry , . *IEEE Transactions and Neural Networks*, 11, 690- 696.

- [11] M. Rowe, Mining User Development Signals for Online Community Churner , ACM Transactions on Embedded Computing Systems, vol. 5, no. 1, 2015.
- [12] Gupta, S., Lehmann, D.R., Stuart, J.A., 2004 Valuing customers. , Journal of Marketing Research 41 (1), 7-18.
- [13] Peter Spirtes, 2010 Introduction to Causal Inference by Peter Spirtes. , Journal of Machine Learning Research 11 (2010) 1643-1662 Submitted 2/10; Published 5/10
- [14] Finian Lattimore and Cheng Soon Ong 2018 A Primer on Causal Analysis, arXiv:1806.01488v1 [cs.LG] 5 Jun 2018
- [15] JOHN R.ANDERSON Causal Analysis and Inductive Learning , Proceedings of the Fourth International Workshop on MACHINE LEARNING June 22-25, 1987 University of California, Irvine 1987, Pages 288-299
- [16] PR Rosenbaum and DB Rubin. The central role of the propensity score in observational studies for causal effects. Biometrika, 70(1):41-55, 1983. URL <http://biomet.oxfordjournals.org/content/70/1/41.short>.
- [17] DB Rubin C Estimating causal effects of treatments in randomized and nonrandomized studies. , Journal of educational Psychology, 1974. URL <http://psycnet.apa.org/journals/edu/66/5/688/>.
- [18] DB Rubin. Bayesian inference for causal effects: The role of randomization , The Annals of Statistics, 1978. URL <http://www.jstor.org/stable/2958688>.
- [19] DB Rubin. Causal Inference Using Potential Outcomes. , urnal of the American Statistical Association, 100(469):322-331, mar 2005. ISSN 0162-1459. doi: 10.1198/016214504000001880. URL <http://www.tandfonline.com/doi/abs/10.1198/016214504000001880>.
- [20] DB Rubin. For objective causal inference, design trumps analysis. , The Annals of Applied Statistics, 2(3):808-840,sep 2008. ISSN 1932-6157. doi:10.1214/08-AOAS187. URL <http://projecteuclid.org/euclid.aoas/1223908042>.
- [21] S Wright. Correlation and causation. , PJournal of agricultural research, 1921. URL <http://www.ssc.wisc.edu/soc/class/soc952/Wright/Wright>
- [22] T Haavelmo. The statistical implications of a system of simultaneous equations. , Econometrica, Journal of the Econometric Society, 11(1):1- 12, 1943. URL <http://www.jstor.org/stable/1905714>.
- [23] Judea Pearl. Causality: models, reasoning and inference. , MIT Press, Cambridge, 2000.
- [24] David Heckerman and Dan Geiger. Learning Bayesian networks: a unification for discrete and Gaussian domains. In Philippe Besnard and Steve Hanks, editors, , Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, pages 274-282. Morgan Kaufman, 1995.
- [25] Yimin Huang and Marco Valtorta Identifiability in causal Bayesian networks: A sound and complete algorithm. , In Proceedings of the Twenty-First National Conference on Artificial Intelligence, pages 1149-1154, Edinboro, Scotland, 2006. AAAI Press
- [26] Lawrence M. Rudner and Tahung Liang Automated Essay Scoring Using Bayes' Theorem , THE JOURNAL OF TECHNOLOGY, LEARNING AND ASSESSMENT. VOL 1 NO 2 (2002)
- [27] G. Moser ; J. Zerubia ; S.B. Serpico. SAR amplitude probability density function estimation based on a generalized Gaussian model , IEEE Transactions on Image Processing Volume 15 Issue 6 June-2006.

AN ARTIFICIAL NEURAL NETWORK APPROACH FOR THE CLASSIFICATION OF HUMAN LOWER BACK PAIN

Shubham Sharma and Rene V.Mayorga

Industrial Systems Engineering, University of Regina, Canada

ABSTRACT

In today's world, the problem of lower back pain is one of the fastest growing crucial ailments to deal with. More than half of total population on the earth, suffers from it at least once in a lifetime. Human Lower Back Pain symptoms are commonly categorized as Normal or Abnormal. In order to remedy Human Lower Back Pain, with the growth of technology over the time, many medical methods have been developed to diagnose and cure this pain at its earliest stage possible. This study aims to develop two Machine Learning (M.L.) models which can classify Human Lower Back Pain symptoms in a human body using non-conventional techniques such as Feedforward/Backpropagation Artificial Neural Networks, and Fully Connected Deep Networks. An Automatic Feature Engineering technique is implemented to extract featured data used for the classification. The proposed models are compared with respect to a Support Vector Machine model; considering different performance parameters.

KEYWORDS

Machine Learning, Artificial Neural Networks, Fully Connected Deep Networks, Support Vector Machine, Lower Back Pain, Automatic Feature Engineering technique.

1. INTRODUCTION

According to statistics, near about 80% of adults go through the pain of lower back at some point in their life [1]. Although there has been a noted increase in the technologies and the number of chiropractors to deal with lower back pain; still the ratio of the LBP patients and the chiropractors is quite large. The parts of back involved in this pain are mainly an arrangement of spine, spinal cord, the disc like structure between vertebrates and the ligaments which connects bone to bone.

Lin L. et al [2006], published work named "A Decision Support System for lower back pain diagnosis: uncertainty management and clinical evaluations". This system is a typical web-based system where all the verification and system validation were done using Turing test [2]. Fourney D. et al [2011], presented a review of clinical pathways for lower back pain and case study of the Saskatchewan Spine Pathway. The main motto of this research was to find differences between clinical pathways and clinical guidelines, its example and testing of its success and about SSP [3]. Jenkins H. [2002], presented a paper in which he mentioned about the classification of low back pain. It described about the different types of lower back pain and its classification using KNN, Logistic Regression, Naïve Bayes, Random Forest, Decision Tree, and CART [4].

In this paper, the Classification Models are generated using different Machine Learning techniques such as: The Support Vector Machine (SVM) method; the Feedforward/Backpropagation Artificial Neural Networks (ANN) technique; and the Fully Connected Deep Network Algorithm, (FCDNA), [5][6][7]. Training and testing of the above classification models are done using a publicly available dataset [8]. This dataset consists of 13 columns, from which the first 12 columns are commonly termed as pelvic parameters or Range of Motion (ROM) Attributes. The Final column of the dataset indicates whether the first 12 column values or pelvic parameters values are Normal back pain or Abnormal back pain symptoms. Range of Motion (ROM) Attributes contained in the dataset are named as follows:
Attribute Label Attribute Name

Table 1. Range of Motion (ROM) Attributes

Attribute Label	Attribute Name
Col1	Pelvic Incidence
Col2	Pelvic Tilt
Col3	Lumbar Lordosis Angle
Col4	Sacral Slope
Col5	Pelvic Radius
Col6	Degree Spondylolisthesis
Col7	Pelvic Slope
Col8	Direct Tilt
Col9	Thoracic Slope
Col10	Cervical Tilt
Col11	Sacrum Angle
Col12	Scoliosis Slope

This paper explains the application of various Machine Learning techniques to correctly classify lower back pain symptoms using featured data obtained by implementing the Automatic Feature Engineering technique on the complete dataset. Section 2 explains the methodology which includes data preprocessing, model generation, and performance analysis. Section 3 explains the results after comparing various models and Section 4 presents some Conclusions.

2. METHODOLOGY

2.1. Data Preprocessing

Original dataset is retrieved from a website named Kaggle [8]. Firstly, the dataset normalized and reordered in Waikato Environment [9]. Automatic Feature Engineering technique is applied on the normalized to extract the features. Preprocessing of data can be shown as below.

Featured columns for this dataset after data reduction process are [col1], [col2], [col3], [col4], [col5], [col5/col10] [col6], sqrt[col5/col10]. These featured data columns are the inputs to the classification models.

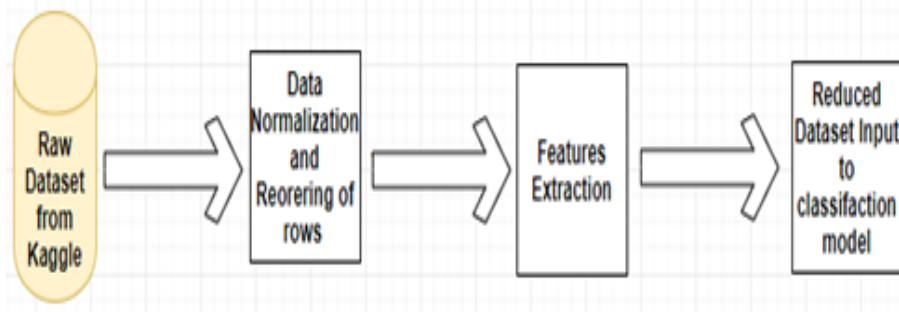


Fig. 1 Data Collection and Preprocessing

2.2. Model Building

Three different classification models are built in this research. Simulation of these models are performed in Rapid Miner Software [10]. This software provides a Graphical User Interface (GUI) for the analytical workflows or commonly termed as process. A process consists of a combination of different operators where each operator is required to do a specific task [10].

The Proposed Artificial Neural Network Classification Model has 2 hidden layers having 8 and 5 neurons in layer 1 and layer 2 respectively. The Activation function is Rectifier and Number of epochs are 100. Stochastic gradient descent (SGD) method is used here to minimize the loss function [11].

The Deep learning Classification Model designed in this research consists of three fully connected hidden layers containing 18, 80 and 2 neurons in layer 1,2 and 3 respectively; with Activation function ReLU (Rectifier Linear Unit) in first 2 and SoftMax in last layer [12][13]. The SGD method is used here to minimize the loss function.

The Support Vector Machine model is using C-SVC type SVM structure, Kernel type: Rbf, Gamma Value: 1.000000000000007, C value:100 and epsilon:0.001 [14].

2.3. Training and Testing

The dataset consists of 311 datapoints out of which 187 datapoints (60% of dataset) are used for training of the classifiers and rest 124 datapoints (40% of dataset) are used for the testing of the trained models and the performance of the models is compared on the basis of performance parameters defined in next section.

2.4. Performance Evaluation Parameters

Performance of the proposed Machine Learning based Classification models are evaluated on the parameters such as Accuracy, Precision, F-measure, Sensitivity, Specificity and Area under Curve (AUC) whose formulae are listed below.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall (or Sensitivity)} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP})$$

$$\text{F-measure} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

[TN- True Negative, TP- True Positive, FN- False Negative, FP- False Positive] ..[15].

AUC is used to know which model predicts the best classification. In this curve, Positive Instances are plotted against negative instances and results are compared [16].

3. RESULTS AND COMPARISON

All the above proposed models are designed, and comparison results are shown in table below.

Table 2. Performance Comparison

S. No.	Classification Model	Accuracy (in %)	Precision (in %)	F-measure (in %)	Specificity (in %)	Sensitivity (in %)	AUC (in %)
1	ANN	88.6	90.3	91.8	78.6	93.3	90.1
2	Deep Network	83.9	88.1	88.1	75	88	92.9
3	SVM	80.7	86.9	85.7	69.8	84.8	86.9

The Classification models are compared based on six performance parameters. As clear from table 2, the ANN classification Model gives best accuracy (88.6%) among all, and the Deep Learning model is second best (83.9%). In terms of precision, the ANN gives best result (90.3%) and the Deep Learning model comes after that (88.1%). The ANN is the best model among the three models in terms of Sensitivity and Specificity with a value of 93.3% and 78.6% for respective parameter. In terms of Area Under Curve (AUC), the proposed Deep Learning model gives best figure (92.9%) and the ANN is second best in this case (90.1%).

The computational time for each developed model is depicted in following table.

Table 3. Computational Times

S. No.	Classification Model	Computational Time (in Sec)
1	ANN	7
2	Deep Network	7
3	SVM	4

Weights of attributes for the featured inputs are shown in the following table (Table 4) with respect to each classification model. Weight of an attribute signifies the impact of that input in the classification of the data.

Table 4. Weights of Attributes

S. No.	Featured Input Data	Classification Model		
		ANN	Deep Learning	SVM
1	Col1	0.017	0.449	0.026
2	Col2	0.108	0.089	0.025
3	Col3	0.028	0.027	0.035
4	Col4	0.067	0.063	0.019
5	Col5	0.041	0.05	0.012
6	Col5/Col10	0.055	0.082	0.006
7	Col6	0.45	0.449	0.106
8	sqrt(Col5/Col10)	0.039	0.007	0.037

It is interesting to notice the somewhat unexpected results given by the Fully Connected Deep Network model based on the test data. It appears that the performance results are not satisfactory enough due to an insufficient amount of data to train this model. The Deep Learning model can perform well when the amount of dataset is large to train the model. The SVM algorithm-based classification model does not perform that well either; due to the limitation of the framework used for the simulation. This study successfully classifies Human Lower Back Pain data using different classification models. It is observed that the highest accuracy 88.64% is yielded by ANN model; whereas, the lowest accuracy of 80.7% is obtained from the SVM model.

4. CONCLUSIONS

This paper provides a non-conventional approach to detect lower back pain in a human body. Twelve Range of Motion attributes' values can decide the type of lower back pain; which is either Normal or Abnormal. Some Machine Learning techniques such as: ANN, Deep learning, and the SVM are used to generate classification models which give promising results. The ANN model gives the best results in terms of Accuracy, Precision, F-measure, Sensitivity and Specificity; while the Deep Learning model is the best model in terms of AUC. Hence, for the practical use, it can be concluded that the proposed the ANN based Classification model can be considered as the "best" model among the three models. Therefore, the ANN model can be used as Clinical Decision Support System (CDSS) by the physicians or chiropractors. The Deep learning model's lower accuracy can be explained by the low amount of available data used in this study. Future work can be done in this field to increase the performance of classification models. Recommendation includes large dataset to train and test the models, trying different algorithms with fine tuning, and other different frameworks to implement this study.

ACKNOWLEDGEMENT

This paper research has been supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

- [1] Rubin, D. I. (2007). Epidemiology and risk factors for spine pain. *Neurologic clinics*, 25(2), 353-371.

- [2] Lin, Lin & Jen-Hwa Hu, Paul & Sheng, Olivia. (2006). A decision support system for lower back pain diagnosis: Uncertainty management and clinical evaluations. *Decision Support Systems*. 42. 1152-1169. 10.1016/j.dss.2005.10.007.
- [3] Fourney, D. R., Dettori, J. R., Hall, H., Härtl, R., McGirt, M. J., & Daubs, M. D. (2011). A systematic review of clinical pathways for lower back pain and introduction of the Saskatchewan Spine Pathway. *Spine*, 36, S164-S171.
- [4] Jenkins, H. (2002). Classification of low back pain. *Australasian Chiropractic & Osteopathy*, 10(2), 91.
- [5] Lau, Suki (10 July 2017). "A Walkthrough of Convolutional Neural Network — Hyperparameter Tuning". *Medium*. Retrieved 23 August 2019.
- [6] TensorFlow for Deep Learning. (n.d.). Retrieved August 6, 2019, from <<https://learning.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>>.
- [7] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". *Machine Learning*. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018.
- [8] Lower Back Pain symptoms dataset, Version 1. Retrieved on May 17, 2019 from <<https://www.kaggle.com/sammy123/lower-back-pain-symptoms-dataset>>.
- [9] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition.
- [10] GmbH, R. (n.d.). Fast Large Margin (RapidMiner Studio Core). Retrieved July 3, 2019, from <https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/support_vector_machines/fast_large_margin.html>.
- [11] Taddy, Matt (2019). "Stochastic Gradient Descent". *Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions*. New York: McGraw-Hill. pp. 303–307. ISBN 978-1-260-45277-8.
- [12] Brownlee, J. (2019 7). A Gentle Introduction to the Rectified Linear Unit (ReLU). Retrieved 27, 2019, from <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
- [13] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). "6.2.2.3 Softmax Units for Multinoulli Output Distributions". *Deep Learning*. MIT Press. pp. 180–184. ISBN 978-0-26203561-3.
- [14] RBF SVM parameters. (n.d.). Retrieved August 27, 2019, from <https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html>.
- [15] Sunasra, M. (2019, February 28). Performance Metrics for Classification problems in Machine Learning. Retrieved August 8, 2019, from <<https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>>.
- [16] Classification: ROC Curve and AUC | Machine Learning Crash Course. (n.d.). Retrieved September 8, 2019, from <<https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>>.

COGNITIVE CITIES AN ARCHITECTURAL FRAMEWORK FOR THE CITIES OF THE FUTURE

Cristiana Carvalho¹, Filipe Cabral Pinto¹, Isabel Borges¹, Gonçalo Machado¹ and Ilídio Oliveira²

¹Altice Labs, Aveiro, Portugal

²Departamento de Eletrónica, Telecomunicações e Informática, University of Aveiro, Aveiro, Portugal

ABSTRACT

Digital transformation has changed management models in cities. The use of tools supported by information and communication technologies has facilitated the planning and control of the urban space allowing a rapprochement between the city and the citizens. This proximity is exponentiated with the advent of the Internet of things becoming possible to permanently know the state of the city and to act on the different infrastructures in a dynamic way. This paper proposes the use of Machine Learning techniques to enhance city management by predicting behaviours and automatically adapt rules mechanisms in order to mitigate city problems contributing to the improvement of lives living or visiting municipalities.

KEYWORDS

Architecture, Learning City, Smart Cities, Machine Learning, Big Data

1. INTRODUCTION

Demographic evolution mirrors the changes felt in cities: on one hand, large cities are receiving more and more people; on the other, small cities tend to be empty of people becoming increasingly isolated and aged. But regardless of the city's demographics, they all share the same ambitions: to evolve by protecting natural resources, optimizing infrastructure and equipment utilization, improving the day-to-day processes and offering more and better digital services leading all to a better way of life in the urban space.

Cities are becoming more and more digital supported by new communication and information technologies and many cities try to be Smart Cities and obtain their benefits [12]. The advent of Internet of Things (IoT) is another step taken towards the automation of internal flows contributing greatly to the city's operational efficiency and cost reduction. The information gathered by sensors is used to know the state of the city and to act on the different infrastructures, contributing to the improvement of the city in all its dimensions [13][14]. But the increasing complexity brought by the addition of more and heterogeneous data sources makes it difficult to configure rules in city analysis systems.

Thus, the logic of the city must be supported by learning mechanisms based on the construction of appropriated prediction models, enabling almost instantaneous answers to the city requests.

This article presents an architecture that uses Machine Learning techniques to predict the occurrence of situations and to introduce contextual recommendations that help to improve the city management. This architecture foresees to change models whenever the dynamics of urban space will require, enhancing the adaptation of the city to the real needs of the citizen.

2. MOTIVATION

The main motivation of this work is the definition of an architecture enabling the management and automation of a city. An urban platform integrating intelligence will allow the city to adapt to current and future needs of the population. Through Machine Learning techniques it is possible to predict behaviors and automate rules mechanisms in order to improve the quality of life of the population in cities.

In this work it will be proposed a future-proof architecture for future cities, combining Smart Cities, IoT and Machine Learning.

3. STATE OF ART

3.1. Smart Cities

Smart City is an expression that is widely used nowadays but its meaning can be analyzed from different perspectives and as so multiple definitions. Despite this diversity, it is generally understood that a Smart City is a city where efficiency and optimization of resources, infrastructures and services are enhanced by the use of technology. The city, according to Sotiris Zygiaris, can be broken down into different pillars: economy, mobility, environment, life and government [1].

Also, [2] defines and establishes methodologies for a set of indicators enabling to measure city services performance and by that, the quality of life of citizens, that are defined in ISO 37120:2018. In [2] are defined 17 themes - Economy; Education; Energy; Environment; Finance; Governance; Health; Recreation; Safety; Shelter; Solid Waste; Telecommunications; Transportation; Urban Planning; Wastewater; Water & Sanitation and finally Fire & Emergency; Response – that are accountable for city quality of life.

Looking to all above domains, the task to improve urban performance is huge. It is essential that city transformation is supported by intelligence and to do so, the city must use all the necessary data from all available of sources together to monitor, analyze and act over the urban space, increasing the collaboration between different economic agents and encourage innovative business models, both in private and public sectors, with the ultimate goal of improving the living standards of the citizen [3].

3.2. Internet of Things

IoT (Internet of Things) "extends" the Internet to objects and leverage a myriad of new services. It will impact on innovation, create new businesses and revolutionize the way we live in society. But IoT is also a new technological paradigm characterized by the availability of a network of machines that are able to communicate with each other [4], bringing things, with computational and communication capabilities to the Internet. In this context, it is necessary to invest efforts in doing the appropriate planning, so that current and future cities are prepared for the necessary development in a robust, creative and sustainable way for the improvement of the society quality of life. IoT is definitely the most important tool to consider in the city transformation [5]. This

concept is also a major challenge, not only at conceptual and technological level, because there are several different technologies in the ecosystem, but also at social and political level [6].

In [9], the authors refers that “*the IoT Application scenarios extend to personal health monitoring, home control, smart grid, smart traffic & transportation management, smart environmental monitoring and more. The applications of IoT and M2M communications in the context of smart cities are the particular interest*”. The authors also emphasize the integration into oneM2M standard architecture by IoT framework for mobile crowd sensing. Also, in [11] is mentioned that “*for smart traffic solutions, crowdsourcing could offer real time traffic update enabling drivers to change the route to destination if necessary*” based in M2M devices.

M2M facilitate the creation of new use cases in different industrial sectors [15] and the deployment of M2M technologies is increasing in urban environments [16].

3.3. Machine Learning

Machine Learning is an Artificial Intelligence branch that gives systems the ability to learn automatically. Its application focuses on the development of models envisaged to learn with data that has been previously collected and processed [7].

All smart cities are digital cities, but not all digital cities are smart cities. While digital cities are enable the provision of services through digital channels, in intelligent cities it is possible to go further by planning, operating and performing all over the urban space in an orchestrated and autonomous way [8].

Learning is a fundamental part of cities’ evolution and the big data help this learning [12]. As more forecasting capabilities are in place as more events can be anticipated and mitigated ensuring the safety and quality of life of citizens. Allowing cities to learn, using the historical data provided by the urban ecosystem makes it possible to speed up the resolution of various problems and improve processes and procedures. Besides historical data, the use of real-time data to feed city brain will enable just in time decision making and autonomous actuation.

4. ARCHITECTURE

This article aims at describing the key functionalities of the proposed architecture in order to facilitate the everyday life in cities. The architecture of cognitive cities encompasses two distinct functional blocks that complement each other. The first one, the city learning model, includes all the necessary components for the construction of the model of Machine Learning to be adopted by the city as well as the permanent evaluation of the deployed one. The second block, the city runtime, is constituted by the necessary elements for the daily management of the city, allowing to know and to act in the infrastructures and equipments taking into account the context of the urban space. Both functional entities resort to a data management platform to store and persist the raw and analyzed city data. Figure 1 presents the cognitive framework for the cities of the future.

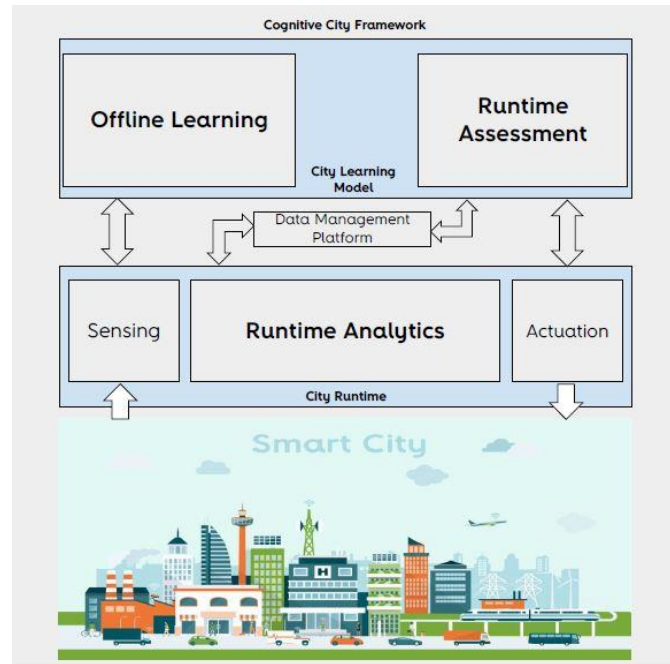


Figure 1. Generic Cognitive City Framework Architecture

4.1. City Learning Model

The city model has the objective of managing the life cycle of the models to be applied in cities. It encompasses the whole learning process, the deploy component in the city's systems, as well as its evaluation taking into account new entries collected by existing urban systems. City model encompasses two major blocks: **offline learning** and **runtime evaluation**. Figure 2 presents the city learning model module.

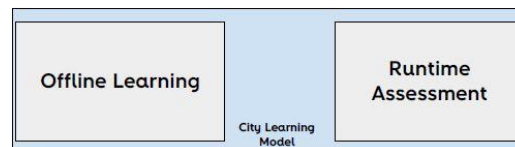


Figure 2. City Learning Model

4.1.1. Offline Learning

The offline learning functional entity includes the key tasks for providing cities with learning capabilities. It is an offline component encompassing all the stages for training the model based on the available dataset of the problem domain. It assumes a data preparation stage where the data preprocessing takes place to arrange the dataset for the Machine Learning algorithms, including checking missing values, dealing with outliers, correcting duplicates, standardizing or splitting data for training and testing. Also, the offline learning includes the proper processes for models creation based on a set of Machine Learning algorithms (Figure 3). The training data is used as input for kNN, Naïve Bayes, Decision Trees, Random Forest, SVM and Neural Networks algorithms resulting in different models able to be applied for prediction.

Each model is evaluated using confusion matrix aiming at determining the performance of the classification models created.

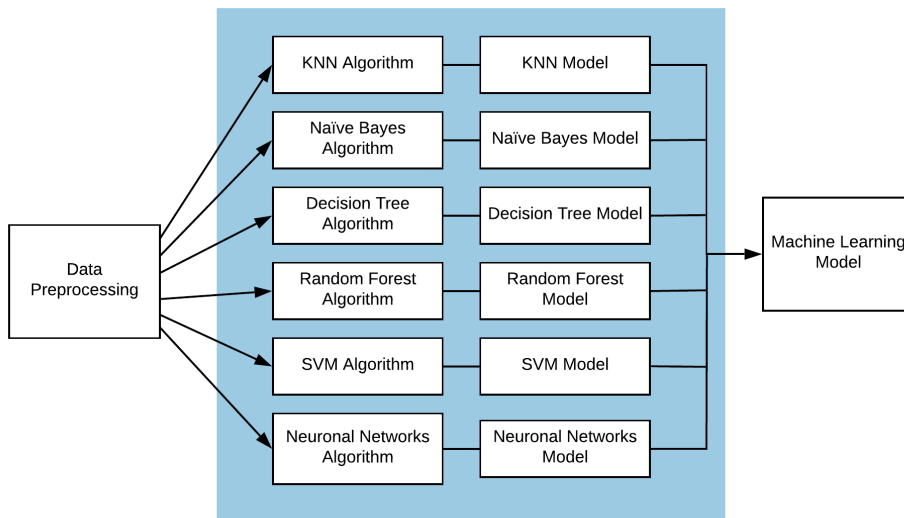


Figure 3. Processes for model creation

Finally, the most appropriate model is selected and deployed in the runtime environment making possible to predict city critical situations and to anticipate mitigation actions. Figure 4 presents the offline learning module.

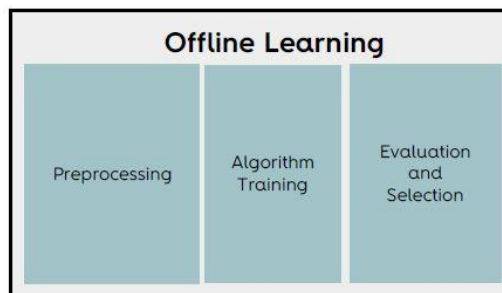


Figure 4. Offline Learning

4.1.2. Runtime Assessment

The runtime assessment aims to ensure that the deployed model remains adapted to the dynamics of the city. For this, it uses the **verification** module to complement the forecast made with the actual result, that is, it verifies whether the forecast done was correct or not. Moreover, it runs the **evaluation** module to verify the model accuracy levels. When the accuracy begins to decrease it is necessary to redo the model taking into account all the new data gathered. Figure 5 presents the runtime assessment module.

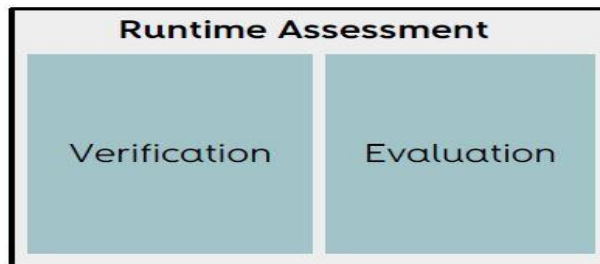


Figure 5. Runtime Assessment

4.2. City Runtime

The city runtime aims at managing the city infrastructure in “real time” in order to make life in cities easier. This module uses the model developed in offline mode to manage occurrences in the city. The data are collected through sensors scattered in the urban space and analyzed in order to obtain a prediction of a certain occurrence. According to the expected result will be made a recommendation of action that will trigger a change in the infrastructures of the city.

The City Runtime encompasses the following components: sensing, runtime analytics and actuation. Figure 6 presents the city runtime module.

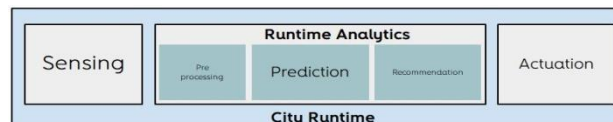


Figure 6. City Runtime

4.2.1. Sensing

Sensing is the entity in charge of collecting physical measurements related with city infrastructure. It resorts to a set of sensors that collect raw data related with different city assets, such as noise, traffic, temperature among others, and transmit it towards the data management platform for storage and further analysis.

4.2.2. Runtime Analytics

The Runtime Analytics is the “brain” of the runtime system. It gets the information sensed and prepares it for the Machine Learning module. The cleaned data sample is used to predict a specific event; it enters in the prediction entity, which runs the Machine Learning model deployed in the city system, allowing getting the classification or regression result. Depending on the prediction outcome, a specific recommendation is set in order to update the city infrastructure status. Figure 7 presents the runtime analytics module.

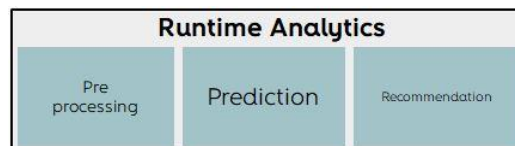


Figure 7. Runtime Analytics

4.2.3. Actuation

The Actuation entity is responsible for the enforcement of the recommendations provided. It closes the loop by sending appropriate commands towards the infrastructure in order to adapt it to the instantaneous city needs.

4.3. City Data Management Platform

The key functionality of the City Data Management Platform is to make data mediation between different system entities. It is a cloud-based open API platform able to upkeep different technologies and protocols facilitating the end-to-end system integration. It supports both request & response and publish & subscribe message exchange patterns. The City Data Management

Platform allows linking different city domains by enabling the storage and the share of city data, solving the typically city information silos issues. All the city data lifecycle is here managed in order to make it permanently available for authorized entities.

4.4. Global Architecture Framework for Cognitive Cities

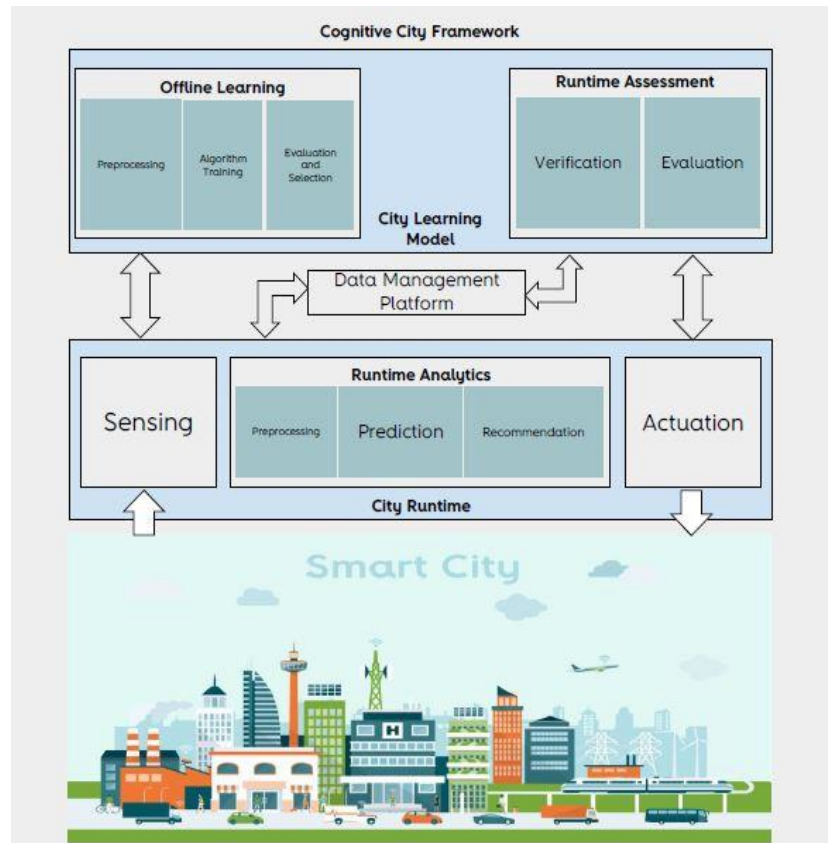


Figure 8. Global Architecture Framework for Cognitive Cities

5. CITY ACCIDENT SCENARIO – AN EXAMPLE

Usually there aren't enough resources for victims' rescue in road accidents. When there are several occurrences at the same period of time, it may be required to manage priorities, enabling to first aid to the most seriously injured. However, it is not always easy, given the huge number of involved factors, to realize quickly which the most severe occurrence was.

António is a citizen of a city that already uses the Cognitive City Framework. Unfortunately, on the way to work on a rainy day, he had an accident. Although slight, medical resources were needed and therefore people who saw the accident called the first aid.

On a road parallel to Maria, she also had an accident, and this was a serious accident and it was urgent that rescue be provided as soon as possible. However, the resources of relief were not many, and there was only one ambulance available. From the information given through the calls, it was not possible to perceive, because of the nervousness of those who spoke, that Antonio's accident despite needing help was not serious. Through various features it is possible to make this prediction and first to rescue Maria.

By applying a Machine Learning model to the attributes that make this prediction possible, it will be possible to prioritize events more effectively.

6. DATA FLOW

6.1. Learning Phase

This phase encompasses all steps required to create a model appropriate to predict specific situations in the city. The City Learning Model block uses the data stored in the Data Management Platform for the modelling process (step 1). Within the Offline Learning entity, the collected data is cleaned in the preprocessing stage resulting in a prepared dataset for model creation (step 2). As can be seen in Figure 8, the dataset works as input for different Machine Learning algorithms, such as, Naïve Bayes, Random Forest, neural networks, among others. For each algorithm, a model is created becoming ready for assessment (step 3). Each model is tested and evaluated, based on performance, and the selected one is deployed in the City Runtime block for runtime usage (step 4).

6.1. Runtime Phase

The Runtime Phase controls the city infrastructure in a dynamic way. It impacts the citizens' daily life by adapting in anticipation the urban space to the city needs. It resorts to the Machine Learning model built to predict city situations and mitigate them in advance.

The sensors spread in the city are periodically making available city data. This data is collected and set into a common information model (step 5), being then sent to the Data management Platform (step 6), which stores and persists the information. The Data Management Platform notifies the Preprocessing module in the Runtime Analytics (step 7), which cleans the data making it ready for the Prediction module. The Prediction runs the model deployed; based on the input it labels the class allowing anticipating critical situations. The prediction done is published in the Data Management Analytics (step 8), which, by its turn, notifies the Recommendation module (step 9). Based on the prediction received, the Recommendation sets the actions required to mitigate the issue and sends it to the Actuation module (step 10). Finally, the Actuation changes the infrastructure state (step 11) adapting the city to the dynamics of daily life.

6.1. Evaluation Phase

This module is responsible for the verification of the veracity of the prediction as well as for the evaluation of the deployed model. In a supervised way, it is checked if the prediction done was correct or not (step 12), being the result stored in the Data Management Platform. Periodically, the Evaluation module gathers all the information (step 13) and compares the prediction with the verified values in order to check the model accuracy. This result is published in the platform (step 14), which notifies the Offline Learning about the new results (step 15). Depending on the system policies, the result of the model evaluation can trigger the creation of a new model to be deployed in the city fitting the citizens' needs.

Depending on the scenario, we need evaluate metrics as accuracy, precision, recall and confusion matrix. If in the scenario is important eliminate false negatives so the most important metric is recall. This metric indicates how well the model identifies positive cases correctly [10]. For example, in sick patient detection, the cost associated with false negative can be very high to the patient. It's important analyze the confusion matrix too. If in the scenario are important predict the most cases correctly so it's important analyze accuracy and precision. The better results are when metrics are close to 1.

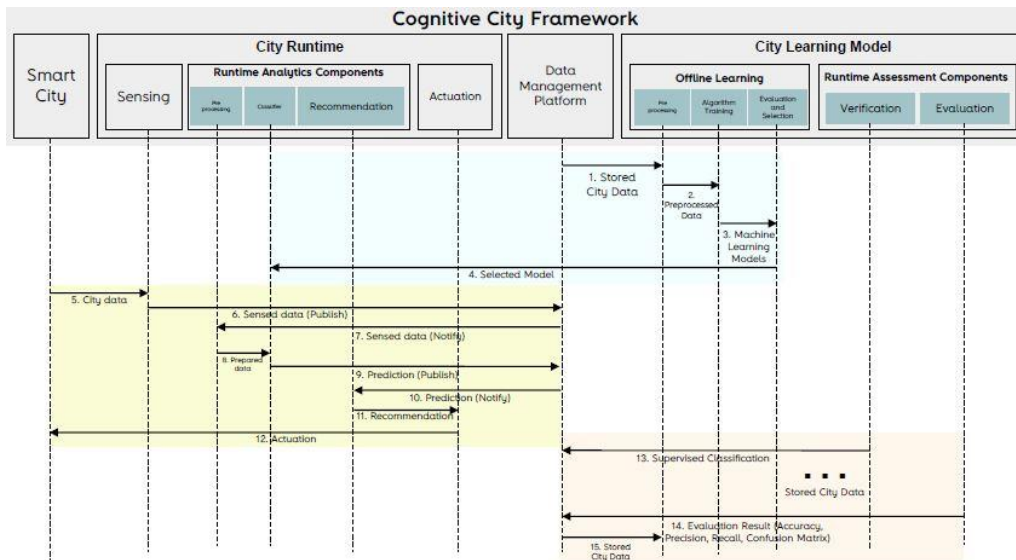


Figure 9. Cognitive City Framework Data Flow

7. CONCLUSIONS

The future of smart cities goes through the Machine Learning applied to them.

The relationship between smart cities, IoT and Machine Learning is quite close and thus they all complement themselves.

By establishing a connection between technology and the majority of people, smart cities may become a way to accomplish a balance on the life quality of the population [4]. The Learning City Framework seeks to facilitate the introduction of Machine Learning in the already existent smart cities.

In the future we will apply the Learning City Framework to the scenario in order to obtain results that prove the quality of this.

ACKNOWLEDGEMENTS

This research is partially a result of the CityAction project CENTRO-01-0247-FEDER-017711, supported by Centro Portugal Regional Operational Programme (CENTRO 2020), under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

REFERENCES

- [1] S. Zygiaris, “Smart City Reference Model: Assisting Planners to Conceptualize the Building of Smart City Innovation Ecosystems,” *Journal of the Knowledge Economy* 4: 2 (2013) 217–231, 2012.
- [2] M.-L. Marsal-Llacuna, J. Colomer-Llinàs, and J. Meléndez-Frigola, “Lessons in urban monitoring taken from sustainable and livable cities to better address the Smart Cities initiative”, 2014.
- [3] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” pp. 431–440, 2015.

- [4] M. R. Santiago and J. V. Payao, "INTERNET OF THINGS AND SMART CITIES: TECHNOLOGY, INNOVATION AND THE PARADIGM OF SUSTAINABLE DEVELOPMENT," pp. 787–805, 2018.
- [5] M. Mancini, "Internet das Coisas: História, Conceitos, Aplicações e Desafios," 2018.
- [6] E. System, "What is Machine Learning? A definition." [Online]. Available: <https://www.expertsystem.com/machine-learning-definition/> [Accessed: 2019-02-01].
- [7] N. Komninos, *Intelligent Cities: Innovation, Knowledge Systems, and Digital Spaces*. Spon Press, 2002. [Online]. Available: <https://books.google.pt/books?id=psQq2PJP07gC>
- [8] D. Evans, "A Internet das Coisas Como a próxima evolução da Internet está mudando tudo," pp. 1–13, 2011.
- [9] S. K. Datta, R. P. Ferreira da Costa, C. Bonnet and J. Härrri, "oneM2M architecture based IoT framework for mobile crowd sensing in smart cities," 2016 European Conference on Networks and Communications (EuCNC), Athens, 2016, pp. 168-173.
- [10] Scikit-learn, "Recall_score." [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.metrics.recall_score.html.
- [11] Datta, Soumya Kanti & Bonnet, Christian. (2015). "Internet of Things and M2M Communications as Enablers of Smart City Initiatives". 393-398. 10.1109/NGMAST.2015.10.
- [12] Al Nuaimi, Eiman & Al Neyadi, Hind & Mohamed, Nader & Al-Jaroodi, Jameela. (2015). "Applications of big data to smart cities."
- [13] Elvira Ismagilova & Laurie Hughes & Yogesh K. Dwivedi & K. Ravi Raman. (2019). "Smart cities: Advances in research — An information systems perspective", *International Journal of Information Management*.
- [14] Eduardo Felipe Zambom Santana, Ana Paula Chaves, Marco Aurelio Gerosa, Fabio Kon, and Dejan S. Milojevic. (2017). *Software Platforms for Smart Cities: Concepts, Requirements, Challenges, and a Unified Reference Architecture*. *ACM Comput. Surv.* 50, 6, Article 78 (November 2017), 37 pages. DOI: <https://doi.org/10.1145/3124391>
- [15] F. Cabral Pinto, P. Chainho, N. Pássaros, F. Santiago, D. Corujo & D. Gomes. (2013). "The business of things architecture."
- [16] I. Vilajosana & M. Dohler, (2015) "19 - Machine-to-machine (M2M) communications for smart cities", *Machine-to-machine (M2M) Communications*, Pages 355-373, ISBN 9781782421023.

AUTOMATED MUSIC MAKING WITH RECURRENT NEURAL NETWORK

You Peng¹, Ariel Jiang² and Qi Lu³

¹Department of Computer Science

California State Polytechnic University, Pomona, CA, 91768

²Department of Computer Science University of California, Irvine

³Department of Social Science University of California, Irvine
Irvine, CA, 92697

ABSTRACT

Today, the growing market of entertainment has placed a higher demand for music. Quality music is essential for video making, video game making, or even in any public places. However, sometimes finding a suitable list of music can be hard and expensive. This may be solved by automatic, deep-learning based music making. Using Recurrent Neural Network, computers are able to learn the patterns from existing music pieces and convert them to a possibility map. Companies like Google, Sony, and Amper are creating their applications for music generation. We plan to set up a platform where generating music can be done and retrieved directly online. With different options for genre and length, the users can conveniently generate music that fits their needs.

KEYWORDS

Music Generation, Machine Learning, RNN, Web Service

1. INTRODUCTION

The traditional definition of music is similar to what is described as: a set of notes that correlate, harmonize, and express emotion. Most classical music [1] was composed of sophisticated structure and arrangement to better convey ideas and emotions. However, modern music [2] has evolved to serve as more of a leisure-time entertainment with a more rhythmic, patterned, and less complex style which may be learned and generated with machine learning [3][4].

While many companies are doing such businesses: for example, Amper, which provides enterprise-oriented music generation service, they are designed to serve large enterprises and are quite expensive for an individual or small group. On top of that, their registration processes are long and complex which limits accessibility. The purpose of the musicgen System [5] is to provide convenient and quick access to AI [6][7] generated music that is open to everyone as well as testing models to test out potential ways to improve the quality of AI-generated music. The system consists of two parts: the front web page and the core code to generate music.

The front page is created using bootstrap [8]. The layout is simple; every parameter is put into a single block to be selected, and the user just have to go from top to bottom to finish selecting. After clicking the button “generate now”, a playable music file will be returned at the bottommost to be previewed or downloaded.

The core code is implemented with Python using Google's open source machine learning library magenta [9]. We trained models with hundreds of sample music online categorized into 4 genres: piano, drums, rock, and electrical. Then, we used flask to retrieve user request and input it into the core code and return it to the user. Finally, we built the website on Amazon's AWS cloud server [10].

2. CHALLENGES

2.1. Music is Complex.

Music is intrinsically complex. It composes of various of elements such as pitches, harmony, rhythm, tempo, etc. It cannot be easily expressed with some combinations of known rules or formula. Unlike taught computers to play chess or Go, where there are clear and fixed rules, it is extremely difficult for a computer to evaluate how well a piece of music is. Therefore, it is difficult to define an objective function to optimize for. The variety and subjectivity of music makes automatic music generation a very challenging task.

2.2. How to build an automatic music generation system that is user-friendly

There have been many approaches for algorithmic music generation, which uses statistical models or artificial neural network. However, these mathematical concepts are difficult for a mature music fans to understand. It is unrealistic to require users to understand how the number of hidden layers and the choice of activation function will affect generated music. It remains a challenge how to enable users to create music with high-level control parameters such as style and rhythm instead of low-level parameters such as number of hidden layer and units, activation function.

3. SOLUTION

In this section, we describe the architecture of our auto music generation system [11] and the algorithms behind that powers this system.

3.1. Overview of the Solution

We have built a web-based [12] interactive music generation system. Figure 1 shows the architecture of the system. For the front-end, we built a web page in html and Javascript. The web UI [13] is shown in Figure 2. It allows users to specify the genres, types and length of generated music. The web page will make Ajax call to the backend server. For the backend, we use Flask to build a server and execute music generation scripts. For the AI part, we use magenta library. Magenta is an open source Python library powered by TensorFlow [14]. It uses Recurrent neural network (RNN) [15] to generate music in MIDI format. The generated music can be played from web page or downloaded for processing using GarageBand.

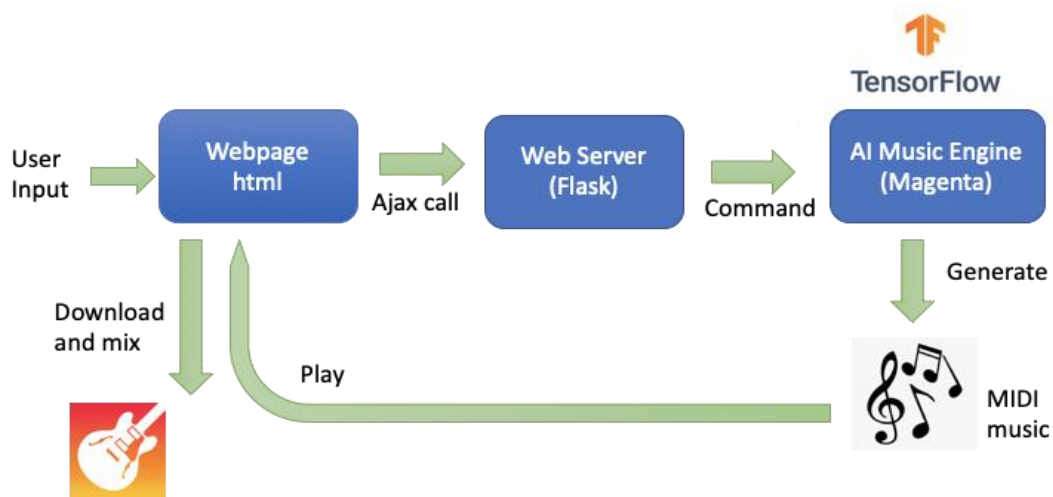


Figure 1. Architecture of auto music generation system

AI Music Generation



Style

Type

Length

Example 1: Playing MIDI files

[Play music](#)

[Stop Playback](#)

MIDIjs status: Initializing ...
 MIDIjs audio time (s): -

Figure 2. Web UI of auto music generation system

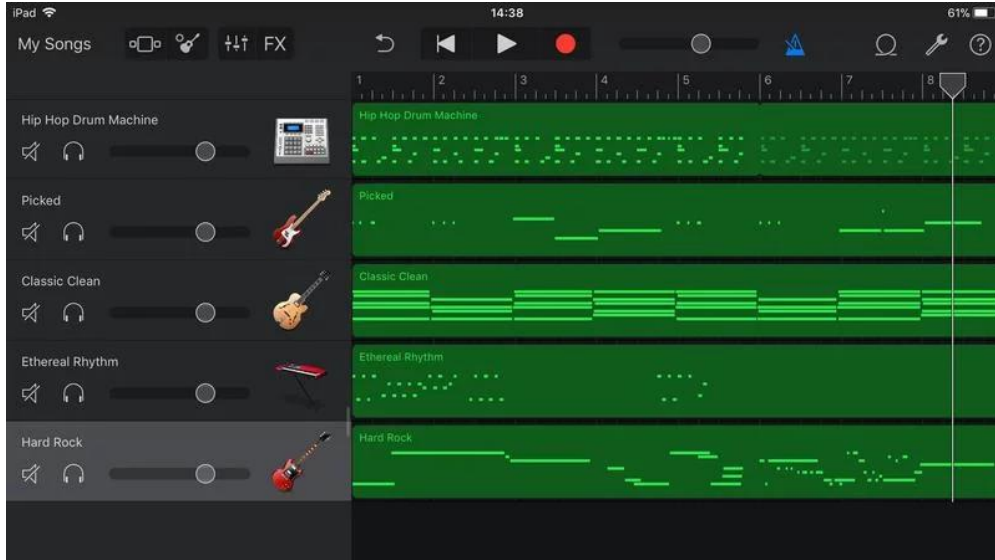


Figure 3. Using GarageBand to mix MIDI music

3.2. Recurrent Neural Network

Recurrent neural network (RNN) is a variant of neural network that can be used for sequential data. Unlike traditional feedforward neural network which only takes a fixed sized input, RNN can take input of arbitrary long. By keeping internal hidden states, the output of RNN not only depends on the input, but also influenced by what it has learnt from the past.

The structure of RNN is illustrated in Figure 4.

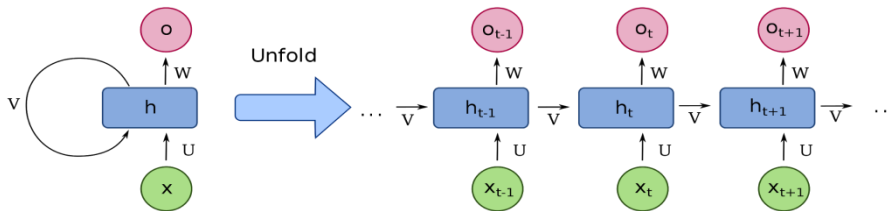


Figure 4: RNN model for auto music generation system

$$h_t = \phi(Wx_t + Uh_{t-1}) \quad (1)$$

Here, the hidden state at time t is h_t . It is a function of the input x_t and hidden state of previous time h_{t-1} . W and U are parameters to be learned. ϕ is the activation function, usually chosen to be sigmoid or tanh.

3.3. Music Generation with RNN

To generate music with RNN, we first need to train the network. By providing different style of training data, the generated music will also be different and conform to the training music. We have used Yamaha e-Piano Competition dataset [16], which contains MIDI captures of ~1400 performances by skilled pianists. Each piece of MIDI is composed of a stream of musical events. An event could be 128 MIDI pitches or velocities. One-hot encoding was used to encode the music into vectors. The RNN network was trained using gradient descent [17]. After the training was completed, the network can be used to generate music similar but not identical to the training music. A starting pitch is needed as the input and RNN will generate subsequent pitches and velocities according to Equation 1.

4. RELATED WORK

There have been persistent efforts for human to develop method to compose music automatically. These efforts can be categorized into different types: some use rule-based system. Some use mathematical models such as fractal, cellular automata or L-system. Boenn [18] used grammars such as harmonics to generate music. Leach et al. [19] analyzed the relationship between fractal and repetitive pattern in music and how to compose music with fractal mathematics. Worth [20] use L-system to generate music. These systems were able to produce reasonable music but the generated music in general lacks variety and novelty.

Recurrent neural networks (RNNs) was first proposed by David [21]. RNNs are able to capture temporal behavior and sequential information. With the increased computational power and algorithm improvement in recent years, RNNs have received great success in many domains such as machine translation [22], speech recognition [23], robot control [24], etc. Also, RNNs have shown surprisingly capability in text generation, such as review generation [25], poetry generation [26].

5. CONCLUSIONS

In this project, we set up a platform where generating music can be done and retrieved directly online, which uses recurrent neural network model to learn the patterns from existing music pieces and convert them to a possibility map. Based on this platform, users can conveniently create their music using proper options for genre and length. Bootstrap is used to develop front page and Google's open source machine learning library magenta is applied to implement RNN model and train it with hundreds of sample music online categorized into several genres.

As for the future work, we will investigate more music genres to update the auto music generation system and make it cover cases as many as possible. Therefore, it can serve users' needs better.

In addition, one limitation related with the auto music generation system is that it does not have enough users in test. we plan to add more features to the system in the next version and collect more feedback from users.

REFERENCES

- [1] Lee, S.hyun. & Kim Mi Na, (2008) "This is my paper", *ABC Transactions on ECE*, Vol. 10, No. 5, pp120-122.
- [2] Goto, Masataka, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. "RWC Music Database: Popular, Classical and Jazz Music Databases." In *Ismir*, vol. 2, pp. 287-288. 2002.
- [3] Adorno, Theodor W. *Philosophy of modern music*. Bloomsbury Publishing, 2007.
- [4] Michie, Donald, David J. Spiegelhalter, and C. C. Taylor. "Machine learning." *Neural and Statistical Classification* 13 (1994).
- [5] Bishop, Christopher M. *Pattern recognition and machine learning*. springer, 2006.
- [6] Eigenfeldt, Arne, Oliver Bown, and Benjamin Casey. "Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble." In *ICCC*, pp. 134-141. 2015.
- [7] Haussler, David. "Quantifying inductive bias: AI learning algorithms and Valiant's learning framework." *Artificial intelligence* 36, no. 2 (1988): 177-221.
- [8] Li, Yuezun, Ming-Ching Chang, and Siwei Lyu. "In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking." *arXiv preprint arXiv:1806.02877* (2018).
- [9] Rubin, Donald B. "The bayesian bootstrap." *The annals of statistics* (1981): 130-134.

- [10] Roberts, Adam, Curtis Hawthorne, and Ian Simon. "Magenta. js: A javascript api for augmenting creativity with deep learning." (2018).
- [11] Talbot, David. "Vulnerability seen in amazon's cloud-computing." *MIT Tech Review* (2009).
- [12] Johanson, Brad, and Riccardo Poli. GP-music: An interactive genetic programming system for music generation with automated fitness raters. University of Birmingham, Cognitive Science Research Centre, 1998.
- [13] Khan, Badrul Huda, ed. *Web-based instruction*. Educational Technology, 1997.
- [14] Offenhartz, John Ken, and Dana Dawes. "Dynamic generated web UI for configuration." U.S. Patent 9,753,747 issued September 5, 2017.
- [15] Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "Tensorflow: A system for large-scale machine learning." In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265-283. 2016.
- [16] Mikolov, Tomáš, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. "Recurrent neural network-based language model." In *Eleventh annual conference of the international speech communication association*. 2010.
- [17] Zou, Daniel Dore Joey. "DJamBot: Music Generation with Music Theory and Dynamics."
- [18] Burges, Christopher, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. "Learning to rank using gradient descent." In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pp. 89-96. 2005.
- [19] Boenn, Georg, et al. "Anton: Answer set programming in the service of music." Proc. 12th Int. Workshop Non-Monotonic Reason. 2008.
- [20] Leach, Jeremy, and John Fitch. "Nature, music, and algorithmic composition." *Computer Music Journal* 19.2 (1995): 23-33.
- [21] Worth, Peter, and Susan Stepney. "Growing music: musical interpretations of L-systems." *Workshops on Applications of Evolutionary Computation*. Springer, Berlin, Heidelberg, 2005.
- [22] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." *Cognitive modeling* 5.3 (1988): 1.
- [23] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.
- [24] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.
- [25] Mayer, Hermann, et al. "A system for robotic heart surgery that learns to tie knots using recurrent neural networks." *Advanced Robotics* 22.13-14 (2008): 1521-1537.
- [26] Yao, Yuanshun, et al. "Automated crowdturfing attacks and defenses in online review systems." *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017.
- [27] Yi, Xiaoyuan, Ruoyu Li, and Maosong Sun. "Generating chinese classical poems with rnn encoder-decoder." *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, Cham, 2017. 211-223.

PREDICTION OF WORKPIECE QUALITY: AN APPLICATION OF MACHINE LEARNING IN MANUFACTURING INDUSTRY

Günther Schuh¹, Paul Scholz², Sebastian Schorr³, Durmus Harman²,
Matthias Möller⁴, Jörg Heib⁴, Dirk Bähre³

¹Laboratory for Machine Tools & Production Engineering (WZL), RWTH Aachen University Aachen, Germany

²Fraunhofer Institute for Production Technology, RWTH Aachen University, Aachen, Germany

³Institute of Production Engineering, Saarland University, Saarbrücken, Germany

⁴Bosch Rexroth AG, Bexbacher Street, Homburg, Germany

ABSTRACT

A significant amount of data is generated and could be utilized in order to improve quality, time, and cost related performance characteristics of the production process. Machine Learning (ML) is considered as a particularly effective method of data processing with the aim of generating usable knowledge from data and therefore becomes increasingly relevant in manufacturing. In this research paper, a technology framework is created that supports solution providers in the development and deployment process of ML applications. This framework is subsequently successfully employed in the development of an ML application for quality prediction in a machining process of Bosch Rexroth AG. For this purpose the 50 most relevant features were extracted out of time series data and used to determine the best ML operation. Extra Tree Regressor (XT) is found to achieve precise predictions with a coefficient of determination (R^2) of constantly over 91% for the considered quality characteristics of a bore of hydraulic valves.

KEYWORDS

Technology Management Framework, Quality Prediction, Machine Learning, Manufacturing, Workpiece Quality

1. INTRODUCTION

Due to the digitalization of manufacturing companies, an increasing number of machine tools are connected – the manufacturing process is mapped digitally [1, 2]. The analysis of the recorded process data allows for measures to improve quality, time, and costs [3]. Therefore, a considerable amount of data is generated, on basis of which decisions often have to be made correctly and in real time [4]. However, large amounts of data and different data sources significantly increase the complexity of the evaluation. Commonly programmed software encounters limitations when processing large amounts of data while facing the high complexity of the application environment. For some problems, however, no algorithm can be used, since an infinite number of scenarios is conceivable and not all variations can be covered rule-based [5, 6]. Given the currently high level of capacity utilization and a thoroughly optimized

production, further efficiency increases are likely to be possible only through the introduction of completely new technologies.

ML is known as the field of study that gives computers the ability to learn without being explicitly programmed and is considered as particularly effective method of data processing with the aim of generating usable knowledge from data [7]. ML systems have the potential to capture complex correlations instantaneously from unstructured data, constantly improve analyses and dynamically adapt to external environmental conditions, which is why they are regarded as particularly promising to further optimize the production process [7-9]. ML applications therefore bear the potential to further increase efficiency in the manufacturing industry and thus ensure the attractiveness of production locations in high-wage countries [4]. ML as a result is becoming increasingly relevant and is being applied to the manufacturing process.

However, there is the challenge that ML as a still rather novel technology in the manufacturing industry is relatively unknown and untested [9]. It is therefore difficult for potential solution providers to fully assess the technology potential, identify relevant applications in the manufacturing industry and thus develop functional solutions to relevant problems. Hence, a framework that supports solution providers in the development process and deployment of ML applications is needed. The first goal of this research paper is to develop a multi-layered structure that can be used to develop ML applications. The second goal is to develop an application for quality forecasting on basis of the multi-layered structure.

Following on from the introduction, Chapter 2. discusses in depth the subject area of ML in manufacturing. In this section, among other things, a framework for applying ML in manufacturing is developed and a use case from Bosch Rexroth AG is described in order to finally select an appropriate path in the framework regarding the specific technology demands of the use case investigated. Chapter 3. deals with the data collection process, the feature extraction, importance, and selection as well as the specific ML operations, their performance results, and evaluation. Chapter 4. concludes this research work and refers to special challenges and future research-related potentials.

2. MACHINE LEARNING IN MANUFACTURING

2.1. Framework for Applying ML in Manufacturing

According to the opinion of SCHUH & SCHOLZ, ML represents the most important among many working areas of artificial intelligence such as e.g. Image Processing and Vision or Natural Language Understanding [9-11]. At this point in time, ML can be interpreted as a rather diverse bundle of technologies. The subject area is still much disorganized. A rather unstructured and partial discussion revolves around different aspects of the technology at different levels [9]. The necessity of the development of such a framework has already been widely demonstrated. Also the different technology layers required for the description of ML applications are derived. Within this research paper the framework will be enriched with more detailed technology information regarding the various decision alternatives per layer as well as a decision sequence along the different layers during the development of ML applications [9]. The framework is designed to provide an easily accessible theoretical overview of the subject area, to accompany researchers as well as solution providers during the implementation process and to support the development of solutions. The framework was developed in vertical reading direction and claims to classify problems from the ML domain into these layers. By the defined processing of the individual layers the selection options are narrowed down more and more, whereby only certain classes and decision alternatives of the following layer are selectable.

Gradually, the decision alternatives of the framework become more granular, which makes the classification of the problem more thorough. A classification according to ML learning strategies is employed because all applications learn in a certain way and specific data types are related to these learning strategies such as e.g. supervised learning, unsupervised learning and reinforcement learning [12]. ML learning strategies are based on the natural learning mechanism of humans [13]. Supervised learning involves learning from examples. Each set of input (features) is labeled with a specific output. After each processed sample, the internal parameters are adjusted (e.g. Neural Networks (NN): weights, Decision Trees (DT): branches, Support Vector Machines (SVM): hyperplane etc.) to minimize a certain error function, often in the sense of the mean square error. In unsupervised learning, only the available data are used to recognize correlations and patterns without defined target classes [13]. Reinforcement learning can shortly be described as the mapping from situations to actions to maximize a scalar reward. The learner is not told which action to take, but instead must discover which action yields to the highest reward by testing [14]. ML tasks such as e.g. classification, regression, and clustering refer to the ML learning strategies and take the specific data formats of the problem into account [12]. Supervised learning can be divided into regression and classification tasks, while unsupervised learning is mainly used for clustering tasks. In the case of regression problems, intervalscaled data are predicted, whereas in the case of classification problems, discrete characteristics such as good or bad can be predicted. The clustering task can be compared with that of the classification, but the target classes are derived from data and are not defined beforehand [13]. ML operations such as e.g. Random Forest (RF) [15], (DT) [15], and (NN) [16] represent the definite procedures which are used for the analysis. They can be described by their internal model structure (e.g. algorithm, hyper parameters, loss function and other general properties). Each ML operation thereby has an individual structure. ML implementation procedures determine which methodical procedure (e.g. KDD (knowledge discovery in databases), Crisp-DM (cross-industry standard process for data mining), and SEMMA (acronym describing the individual steps of the method)) which programming language (e.g. R and Python) and library (e.g. Scikit-Learn, TensorFlow, Keras, and Theano) should be used [17-19]. The development of methodologies, software tools and languages serve as the standardized approach for industrial applications of data processing [20]. Methodical procedures support the systematic implementation of real-life ML applications. The selection of the contents from this layer depends on the use case. For the development of a ML application the layers are processed and defined in the sequence presented here. The well founded examination of the fundamental literature and implemented ML applications allows for an empirical inductive conclusion on the following features per layer and suggests an integration of the layers into the following decision sequence. The wide variety of literature sources presented above support this classification [9-20].

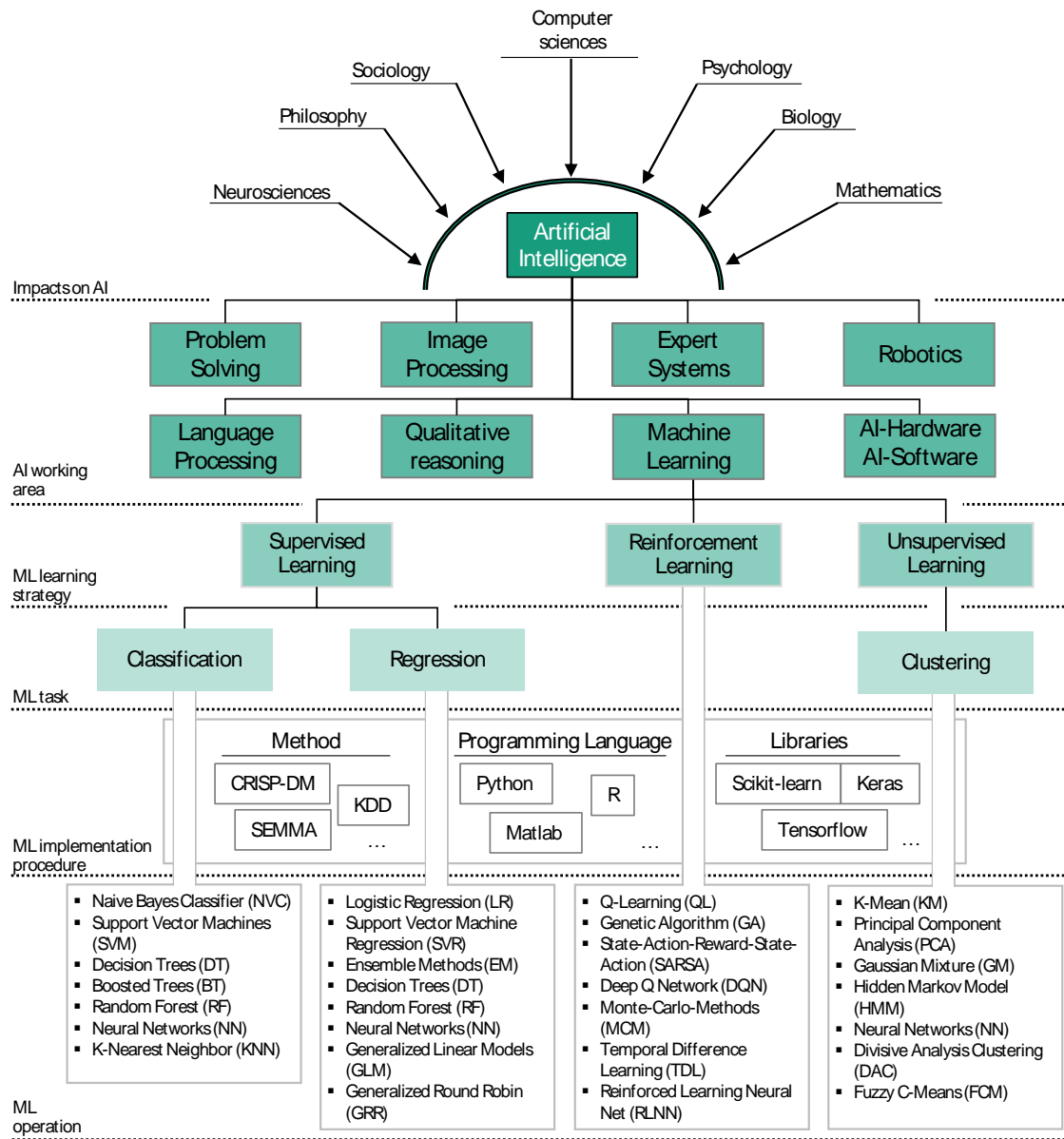


Figure 1. Framework for applying ML in manufacturing

The specific benefit of the framework for the user is that each analysis attempt using ML represents a unique path in the framework. Established paths in the framework can serve as a blueprint for a potential future application design. Thus, less development time may be needed to solve certain problems. In addition, there are more validated and proven solution options to problems, which ultimately leads to lower overall opportunity costs for solution providers.

2.2. Description of the Use Case

The manufacturing process considered within this research is an integral part for the serial production of hydraulic valves at the Bosch Rexroth AG. Hydraulic valves are characterized by bores with tolerances of only a few microns to enable seal-less fits and to prevent oil leakage. A valve is machined first by a milling machine, assembled and finally tested, see Fig. 2. Slight anomalies in the production process can lead to unacceptable quality deviations causing high scrap rates and financial losses. To guarantee the required quality of bores of a valve, sampling

inspection after the machining process (approx. 1% of a batch) and end of line testing (100% of a batch) are applied in industry. Both quality control methods lead to indirect costs and a high latency between the machining of a valve and its measurement results. Therefore, feasible and affordable in-process quality control methods are desired from manufacturing companies. The so gained increased transparency over the machining process is used to make adjustments as soon as the required quality is not reached. Process data from the machining process and ML operations are used to predict the quality of one of the most quality critical bores of a valve.

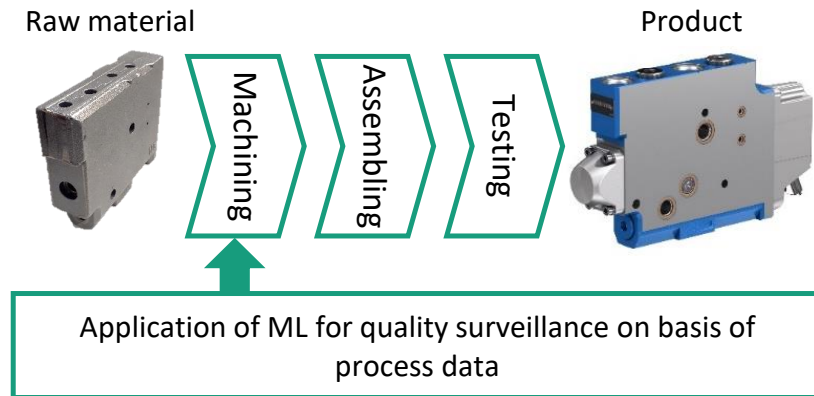


Figure 2. Manufacturing process of a hydraulic valve with quality surveillance

The quality of a bore is determined by dimensional (diameter) and locational (concentricity) quality characteristics. Quality deviations are caused by wear on cutting tools that are used to drill and to ream bores. Tool wear leads to increased cutting forces and torque, which can be measured directly from the drives of a milling machine in form of the motor current and torque [20, 21]. This indirect measurement approach together with ML operations is the only economical technique to obtain quality predictions with a latency close to zero when facing industrial conditions. Such a quality surveillance detects quality deviations in an early manufacturing state and enables cost and resource savings.

2.3. Selection of the Right Path in the Framework Suitable to the Use Case

ML is considered as one working area of AI that can take the manufacturing industry to a next level. In order to cope with the use case considered in this research, technologies from the context of ML should therefore be used. First the learning strategy supervised learning is chosen because the quality of the considered bore of each valve is available and less training data are required to build a model. This leads to a less time-consuming data collection process. In addition, the diameter and concentricity of the bore have to be expressed as numerical values, which defines the ML task as a regression task. For the ML implementation procedure the method CRISP-DM is chosen because an understanding of the business and the data already exists and data preparation is required in form of feature extraction and selection. Due to modelling and evaluation the best ML operation suitable for the use case is determined and finally deployed. The lists of ML operations helps to choose the right category of algorithm regarding a regression ML task. For the implementation the programming language Python and the library Scikit-learn in combination with the distribution Anaconda are used. These choices determine the path in the framework to accomplish the quality prediction of machined workpieces.

3. METHODOLOGY

Fig. 3 summarizes the methodology of this research to obtain a quality prediction for each workpiece from the data of the related machining process. Each process step is described in detail in the following subsections.



Figure 3. Approach to obtain a quality prediction from machining data

3.1. Data Collection Process

The process data were collected during the serial production of hydraulic valves from a milling-machine. To operate a milling-machine a numerical control unit (NC) is used, which processes data from sensors integrated into the machine and the drives. The measured actual values obtained by sensors give a direct feedback on the cutting conditions and are used to control the machine. These machine-internal data were directly collected with a frequency of 1,000Hz from the NC and are the input data for the feature extraction. Selected features are used to make predictions with ML operations. The stored process data were the actual torque values of the drive of the z-axis as well as the spindle, the actual position values of the x-, y-, and z-axis and also the actual speed value of the spindle. Process data from a total of 160 workpieces were collected during the machining. Fig. 4 exemplarily depicts the actual torque value of the spindle for the second tool, which are used to machine the considered bore.

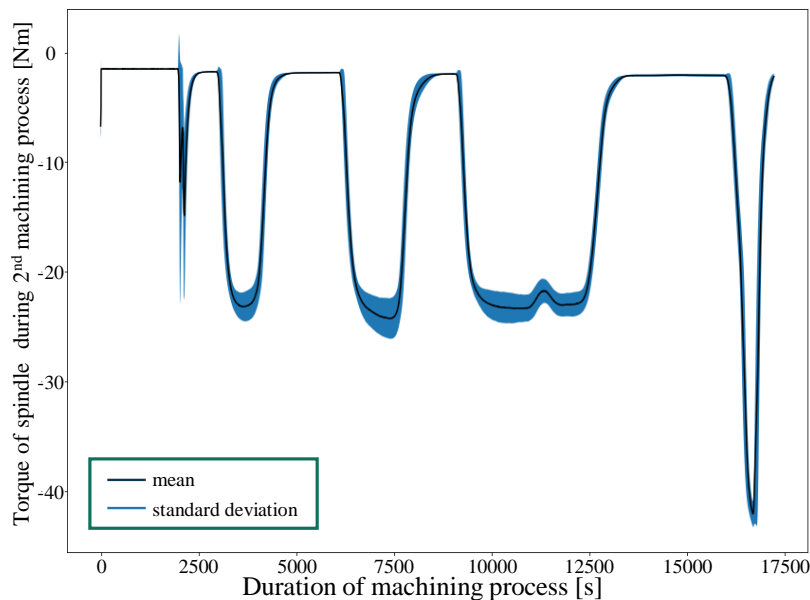


Figure 4. Visualization of process data of the 2nd machining process

In addition, for each workpiece the dimensional and locational quality characteristics were measured representing the target and output parameter for the ML operations. The measured characteristics are the diameter and concentricity of a bore.

3.2. Feature Extraction

To reduce the complexity and volume of the collected time series data as well as to accelerate the training and test process of the ML operations a feature extraction is applied. The aim is to extract characteristics in order to accordingly represent the time series with a reduced set of features. General features (e.g. the shape of the signal) and statistical features (e.g. mean or standard deviation) are very common to represent time series data. However, both forms of features often do not contain all the information needed to expose dynamical time series data. Therefore, VUNUNU ET AL. also analyzed the frequency domain to identify additional patterns in the data [22]. In this paper, 63 methods are used to calculate 794 features for each machining operation with the tsfresh library. For each process parameter a total number of 2382 features is obtained due to the three subprocesses [23]. The extracted feature can be divided into three groups as shown in Fig. 5.

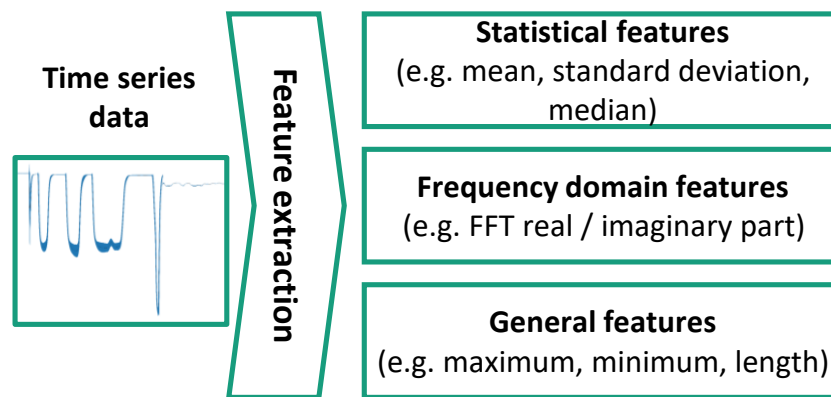


Figure 5. Visualization of process data of the 2nd machining process

3.3. Feature Importance and Selection

Irrelevant features can lead to weak prediction performances and increased computational costs. In a huge set of features the probability of containing irrelevant features is high and therefore a feature selection is recommended. The determined importance of each feature increases the interpretability of a feature in the related use case and enables to select the features that contribute most to an accurate and efficient prediction [24]. In this paper, the RF technique is used to determine the feature's importance. The RF algorithm uses subsets of the training data to construct each of the tree configurations. Therefore, an independent test set is not required for the evaluation of the feature ranking. The feature importance is obtained by comparing the prediction errors after permuting the feature's values of all examples [25]. Feature importance of the RF is well suited for datasets with a small sample and big feature size [26]. The importance of all features is calculated and the features with the lowest importance values are excluded from the data. The features are ranked in descending importance order as shown in Fig. 6. The three most important features are the low frequency fast Fourier transformation (FFT) coefficient of tool 3 (8.97%), the approximate entropy of tool 2 (7.93%) and the high frequency FFT coefficient of tool 2 (6.79%).

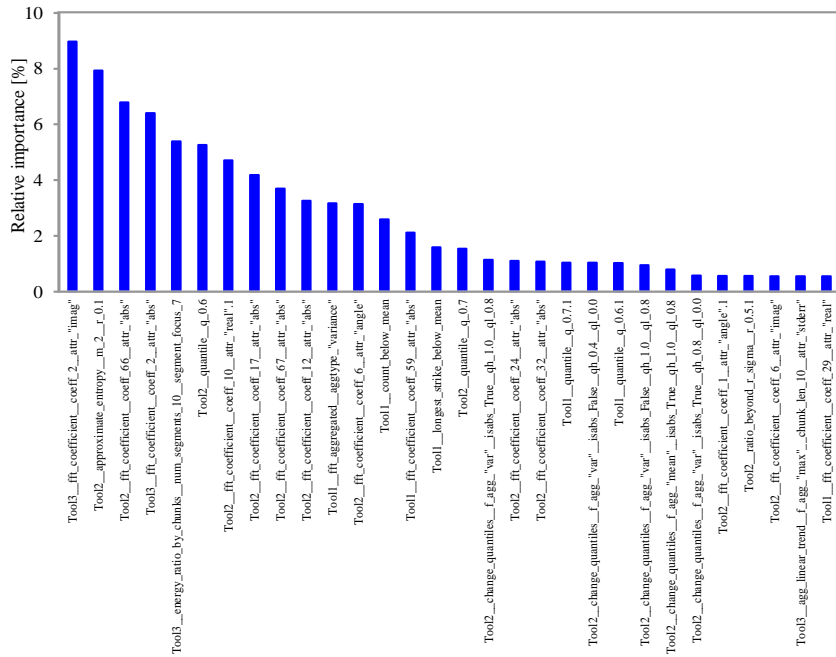


Figure 6. The 30 most important features

To decrease the model complexity and the computational costs the 50 most important features are selected. For the selection of the most relevant features a systematic approach is considered. This procedure answers the question of how many features are needed to maintain the initial accuracy. The initial accuracy is determined by a grid-searched operation based on all features. The individual features ranked by importance are made incrementally available to the operation as a training set. First, the operation is trained with the highest ranked feature. Then, the next important feature is inserted into the feature table and the procedure is repeated. For each feature subset the R^2 is evaluated with a five-fold cross validation. These steps are performed for every feature in the dataset. The mean and the standard deviation of the results for the first 50 important features are shown in Fig. 7.

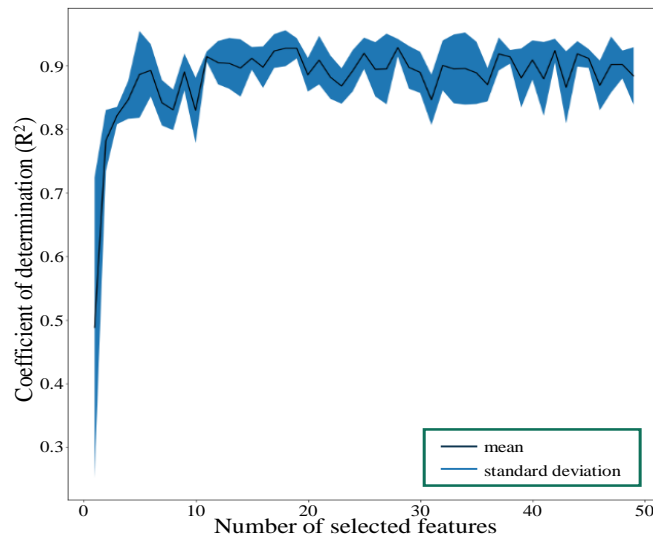


Figure 7. R^2 plotted over the number of selected features

It can be stated that already with the ten most important features a R^2 of 91.39% is reached. Additional features do not lead to an extraordinary increase in accuracy or a saturation behavior, which is possibly due to the small number of samples. By adding more features, the accuracy decreases (not shown here). Thus, the number of features which serve as inputs to the ML operations is set to 50.

3.4. Machine Learning Operation

An artificial neural network (ANN) is a computational model consisting of a collection of artificial neural neurons inspired by the brain of living beings. All those neurons are linked via interconnections to form a large network. Information are insert into the network via input neurons and are conveyed within the network to provide results to the output neurons [27, 28]. The output of each neuron y is determined according to equation (1) by its activation function ϕ that activates the neuron if the weighted sum of n inputs x_i multiplied with the related weights w_i is above a particular threshold u [28].

$$y = \phi \left(\sum_{i=1}^n w_i x_i - u \right) \quad (1)$$

Ensemble methods combine several ML algorithms to achieve a higher prediction accuracy. Often these methods are tree-based like Random Forest (RF) and Extra Trees (XT). Random Forest, developed by BREIMAN, is a ML operation for classification and regression problems [15]. An ensemble of decision trees is used to obtain the final prediction y by averaging the predictions y_t of all trees T for a given dataset x as denoted in equation (2). To build a single tree bagging is applied that describes the random selection of a sample from the original training dataset. RF do not overfit due to bootstrap aggregation, can reach highly accurate predictions, and are fast to train [29, 30]. The ML operation XT differs from RF in two main points to grow the trees. First, XT uses the entire training dataset instead of bagging. Second, splitting-points are chosen fully randomly to split a node [31]. In this paper the RF and XT, as regression tree approaches, are used to predict numerical values for the quality characteristics.

$$y(x) = \frac{1}{T} \sum_{t=1}^T y_t(x) \quad (2)$$

Support Vector Machine (SVM) is a ML operation for classification and regression tasks. Input data are mapped to a high dimensional feature space where the prediction task gets linearly separable. The support vector kernel together with the number of support vectors and its parameter determine the decision function of SVM [32, 33]. Further explanation can be obtained from STEINWART & CHRISTMANN as well as VAPNIK [33, 34].

For the considered ML operations ANN, RF, XT, and SVM the optimal hyperparameters have to be determined. For this purpose, a parameterspace is set up in which different variations of hyperparameters are examined in a grid and the R^2 of each combination is measured. For the individual operations, both the absolute and the repetitive accuracy are calculated. Thus, the best model configuration obtained from the grid search can be determined from the results shown in Table 1.

Table 1. Results of Gridsearch

ML operation	R ² [%] (absolute)	R ² [%] (repetitive)
Artificial Neural Network (ANN)	55.13 ± 5.30	51.63 ± 3.20
Random Forest Regressor (RF)	91.28 ± 2.81	89.59 ± 0.82
Extra Trees Regressor (XT)	93.64 ± 0.65	92.81 ± 0.18
Support Vector Machine Regressor (SVM)	<0	<0

The highest R² can be obtained for the XT Regressor. The associated standard deviation is the lowest of all ML operations, which proves the suitability for this task. The hyperparameters and other properties for the best operation are listed in Table 2.

Table 2. Extra Trees Regressor summary

Parameter	Value
Splitting criterion	MSE
Max. features to consider best split	50
Min. number of samples to split	3
Number of trees	10
Mean fit time	0.115 sec
Mean score time	0.103 sec
R ² of training data	99.95 ± 0.02%
R ² of test data	93.64 ± 0.65%

3.5. Results and Evaluation

The prediction accuracy reached with the described ML operation (XT) is evaluated using the R², the mean squared error (MSE), and the mean absolute error (MAE). Table 3. summarizes the results for the two predicted quality characteristics diameter and concentricity. The diameter is predicted with a higher precision compared to the concentricity. The MAE is with 0.26 μm very little and the MSE is close to zero. For the concentricity the MAE and MSE are much higher than the obtained results for the diameter but the predicted values are still precise enough for the utilization in the described use case.

Table 3. Extra Trees Regressor summary

	Diameter	Concentricity
R ² [%]	93.75	91.66
MSE [mm ²]	1.2x10 ⁻⁷	0.008
MAE [μm]	0.26	22.91

That fact that the desired quality characteristics can be predicted very reliably with XT can be retrieved from the high R² (Table 3.) as well as in the graphical comparison of the predicted and the measured values in Fig. 8.

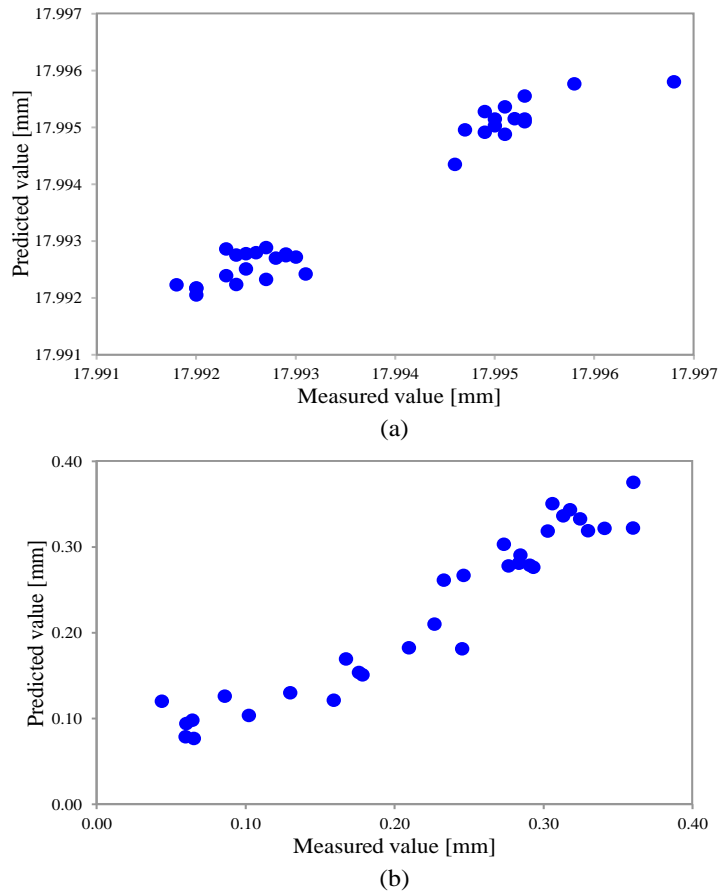


Figure 8. Graphical representation of the predicted and measured values for (a) diameter and (b) concentricity.

To diagnose the systems behavior the learning curve is reviewed. The R^2 is constantly high for the training data, which is typical for an XT. However, a learning effect can still be observed as the R^2 improves with an increasing number of training examples from around 70% to 94%. An R^2 of 100% is hardly achievable, since the learning curve stagnates. In this case a saturation did not occur. Fig. 9 depicts the learning curve. The coefficient of determination (R^2) is plotted with an increasing number of training examples. It can be observed that the accuracy increases with an increasing number of training samples, which proves that the XT is becoming continuously accurate.

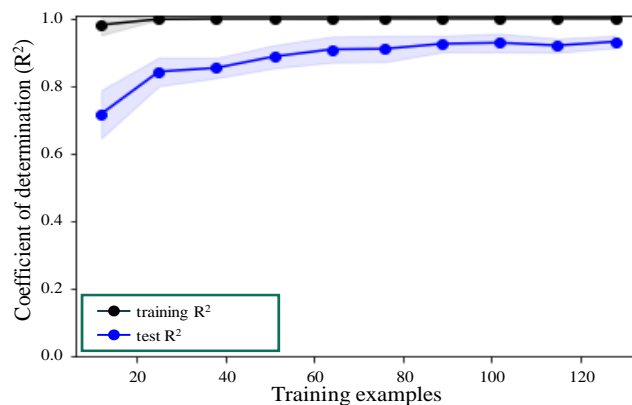


Figure 9. Learning curves of Extra Trees Regressor

4. CONCLUSION

Engineers in the manufacturing industry often have only limited knowledge of using and applying AI or ML to solve problems in their business. Vice versa have data scientists only basic expertise concerning manufacturing processes. The framework presented in this paper enables engineers to gain an overview of AI and ML and its related subfields. The framework guides through the ML implementation process and depicts potential tools and approaches for the application of ML in the manufacturing industry. For a specific use case from the Bosch Rexroth AG the engineers applied the framework successfully. All required decisions were taken along a chosen path in the framework which finally led to a ML operation that predicts the quality characteristics of a bore very precisely. A feature extraction and selection is mandatory to reduce the complexity of the time series data as well as the computing time and to increase the prediction accuracy. From the 794 extracted features, 50 features were determined as significant and were considered for the prediction models. The predictions for the diameter and the concentricity of the considered bore were evaluated with the statistical criterions R^2 , MSE, and MAE. With a MAE of only 0.26 μm for the diameter and 22.91 μm for the concentricity the prediction errors were very low. Hence, XT can be used to predict the quality of bores on basis of process data close to real time. This direct feedback regarding the manufacturing process leads to scrap avoidance, resource savings and cost reductions. Consequently, the application of ML is a further contribution to secure production plants in high-wage countries.

In the future, we will gather more data to increase the training samples for the ML operations in order to improve the prediction accuracy. With more training samples ML operations like ANN or SVM will potentially lead to acceptable results, too. Furthermore, we are looking for further use cases where we can apply the described framework to achieve improvements by implementing ML.

REFERENCES

- [1] X. Yao, J. Zhou, J. Zhang, and C. R. Boer, "From Intelligent Manufacturing to Smart Manufacturing for Industry 4.0 Driven by Next Generation Artificial Intelligence and Further On," in 5th International Conference on Enterprise Systems (ES): IEEE, September 2017, pp. 311–318.
- [2] M. Syafrudin, G. Alfian, N. L. Fitriyani, and J. Rhee, "Performance analysis of IoT-Based sensor, Big Data processing, and Machine Learning model for real-time monitoring system in automotive manufacturing," *Sensors*, vol. 18, 2018.
- [3] E. Alpaydin, *Introduction to Machine Learning*. MIT Press Cambridge, vol. 3, 2014, pp. 1-20.
- [4] McKinsey Global Institute, *Artificial Intelligence. The Next Digital Frontier?*, <https://www.mckinsey.com/~media/McKinsey/Industries/Advanced%20Electronics/Our%20Insights/How%20artificial%20intelligence%20can%20deliver%20real%20value%20to%20companies/MGI-Artificial-Intelligence-Discussion-paper.ashx#page=55>, 2017.
- [5] E. Brynjolfsson and T. Mitchell, "What can machine learning do? Workforce implications," *SCIENCE*, vol. 358, December 2017, pp. 1530-1534.
- [6] A. Agrawal, J. Gans, and A. Goldfarb, *Prediction Machines: The Simple Economics of Artificial Intelligence*. Harvard Business Review Press, 2018.
- [7] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow. Concepts, tools, and techniques to build intelligent systems*, Sebastopol, O'Reilly Media, 2017.

- [8] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, p. 886, 2015.
- [9] G. Schuh and P. Scholz, "Development of a framework for the systematic identification of AI application patterns in the manufacturing industry," 2019, unpublished.
- [10] J. Lunze, *Künstliche Intelligenz für Ingenieure. Methoden zur Lösung ingenieurtechnischer Probleme mit Hilfe von Regeln, logischen Formeln und Bayesnetzen*. Berlin, De Gruyter Oldenbourg, 2016.
- [11] N. J. Nilsson, *Principles of Artificial Intelligence*. Palo Alto, Morgan Kaufmann, 2016.
- [12] J. Brownlee, *A Tour of Machine Learning Algorithms*, <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>, 2013.
- [13] S. Raschka, V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Birmingham: Packt Publishing, 2017.
- [14] R. S. Sutton, *Reinforcement Learning*. New York: Springer Science+Business Media, 1992.
- [15] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.
- [16] G. D. Rey and K. F. Wender, *Neuronale Netze. Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Bern, Huber, 2008.
- [17] U. Shafique and H. Qaiser, "A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA)," *International Journal of Innovation and Scientific Research*, vol. 12, pp. 217–222, 2014.
- [18] G. Piatetsky, Python overtakes R, becomes the leader in Data Science, Machine Learning platforms, <https://www.kdnuggets.com/2017/08/python-overtakes-r-leader-analytics-data-science.html>, 2017.
- [19] S. Shetty, Why TensorFlow always tops machine learning and artificial intelligence tool surveys, <https://hub.packtpub.com/tensorflow-always-tops-machine-learning-artificial-intelligence-tool-surveys/>, 2018.
- [20] F. Klocke, *Manufacturing Processes 1: Cutting*. New York: Springer, 2011.
- [21] M. Eynian, K. Das, A. Wretland, "Effect of tool wear on quality in drilling of titanium alloy Ti6Al4V, Part I: Cutting Forces, Burr Formation, Surface Quality and Defects," *High Speed Machining*, vol. 3, pp. 1-10, 2017.
- [22] C. Vununu, K. Ryong, E. J. Lee, K. S. Moon, S. H. Lee, "Automatic fault diagnosis of drills using artificial neural networks," *16th IEEE International Conference on Machine Learning and Application*, pp. 992-995, 2017.
- [23] M. Christ, N. Braun, J. Neuffer, A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72-77, 2018.
- [24] I. Guyon, A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.
- [25] J. Rogers, S. Gunn, "Identifying feature relevance using a random forest," in: *Subspace, Latent Structure and Feature Selection*. Berlin: Springer-Verlag, pp. 173-184, 2006.

- [26] C. Strobl, A. L. Boulesteix, A. Zeileis, T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, 2007.
- [27] I. N. Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, S. F. R. Alves, *Artificial Neural Networks: A practical course*. Switzerland: Springer International Publishing, 2017.
- [28] H. Rathore, *Mapping Biological Systems to Network Systems*. Switzerland: Springer International Publishing, 2016.
- [29] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, pp. 1063-1095, 2012.
- [30] A. Cutler, D. R. Cutler, J. R. Stevens, *Random Forests*, in: *Ensemble Machine Learning: Methods and Applications*, New York: Springer Science+Business Media, pp. 157-175, 2012.
- [31] P. Geurts, D. Ernst, L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, p. 3-42, 2006.
- [32] L. Zhang, W. Zhou, L. Jiao, "Wavelet Support Vector Machine," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 34, pp. 34-39, 2004.
- [33] I. Steinwart, A. Christmann, *Support Vector Machines*. New York: Springer Science+Business Media, 2008.
- [34] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

A FACIAL RECOGNITION-BASED VIDEO ENCRYPTION APPROACH TO PREVENT FAKEDEEP VIDEOS

Alex Liang¹, Yu Su² and Fangyan Zhang³

¹St. Margaret's Episcopal School, San Juan Capistrano, CA 92675

²Department of Computer Science, California State Polytechnic University,
Pomona, CA, 91768

³ASML, San Jose, CA, 95131

ABSTRACT

Deepfake is a kind of technique which forges video with a certain purpose. It is in urgent demand that one approach can detect if a video is deepfaked or not. It also can reduce a video to be exposed to slanderous deepfakes and content theft. This paper proposes a useful tool which can encrypt and verify a video through proper corresponding algorithms and detect it accurately. Experiment in the paper shows that the tool has realized our goal and we can put it into practice.

KEYWORDS

Video Encryption, Video Verification, Encryption Algorithm, Decryption algorithm

1. INTRODUCTION

Deepfakes [3][4][5] have become a more prevalent problem, with no solution. Not only is there an inability to determine whether or not videos are deepfaked, the amount of people researching deepfake detection are far outnumbered by those researching deepfake synthesis. Deepfakes have real world implications. For example, a video of Gabon's president was decried as a deepfake, nearly resulting in a military coup. deepfakes have begun to extend to America as well, as videos of politicians and influential figures are being falsified, like a video which slows Nancy Pelosi's speech. Some researchers are trying to use A.I. to detect if a video has been edited, others wish to use blockchain to detect deepfakes.

In order to verify digital content for authenticity and avoid slanderous deepfakes [6][7]. Gyfcats has implemented a system in which, after flagging a video under suspicion that it is a deepfake, it combs its database to search for similar videos.

However, software that can spot AI-manipulated videos will only ever provide a partial fix to this problem. As with computer viruses or biological weapons, the threat from deepfakes is now a permanent feature on the landscape. And although it's arguable whether or not deepfakes are a huge danger from a political perspective, they're certainly damaging the lives of women here and now through the spread of fake nudes and pornography.



Figure 1: an example of deepfake (left: real, right: deep faked)

We are trying to do something similar; our goal is to add a mark/encrypted message onto the video before it is released, so we can detect if it has been edited later on. First, we change some pixels to certain values, before utilizing facial-detection to change a pixel, relative to the face. When we are trying to detect, the software will check if the ‘face pixel’ and ‘static pixels’ are the correct color, and if they are, the video is considered ‘real’ [10][11].

Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

To implement this system, there are mainly two challenges that we have to overcome.

2.1. Choose Proper Algorithm to Encrypt

Currently, there are various algorithms can encrypt a video. Considering the efficiency and effects, we should to compare those existing encryption algorithms and choose optional solution. For example, a 2 second video takes around 15 seconds to encrypt. Users can not accept so long time to finish encryption for a large video. When choosing encryption algorithm, we have to consider all aspects, such as accuracy, quality, and efficiency.

2.2. Video Verification

After encrypting a video and finish transferring over internet, how to decrypt and verify the video is equally important. However, facial recognition does not have 100% consistence for encrypting. We have to overcome this challenge and make it more consistent in encryption. In addition, a video usually is compressed when sending out, which may fail in verification due to compression issue.

3. SOLUTION

Figure 2 shows an overview of the proposed approach. First, we change some pixels to certain values, before utilizing facial-detection to change a pixel, relative to the face. When we are trying to detect, the software will check if the ‘face pixel’ and ‘static pixels’ are the correct color, and if they are, the video is considered ‘real’.

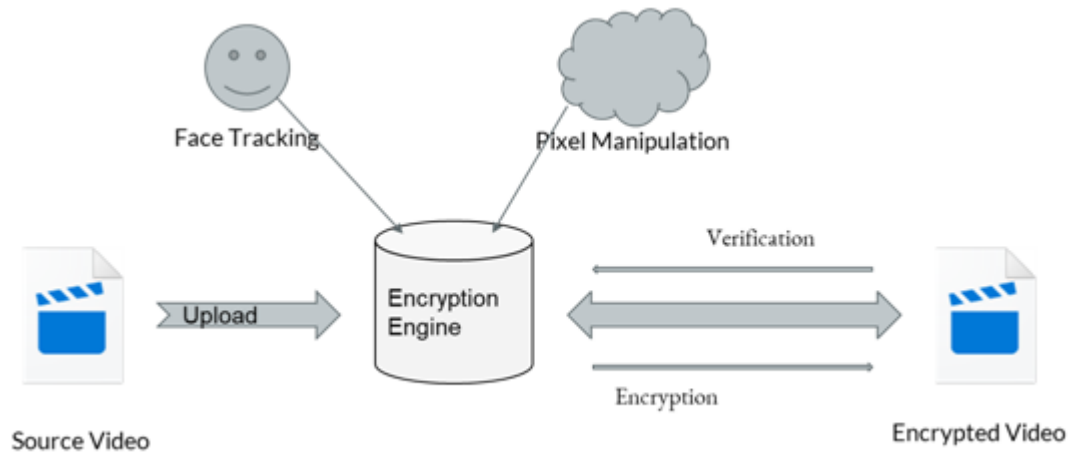


Figure 2: Overview of the solution

3.1. OpenCV

About. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products [8].

To build our face recognition system, we will first perform face detection, extract face embeddings from each face using deep learning, train a face recognition model on the embeddings, and then finally recognize faces in both images and video streams with OpenCV [9].

Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The advantage is that the majority of the picture will return a negative during the first few stages, which means the algorithm won't waste time testing all 6,000 features on it. Instead of taking hours, face detection can now be done in real time.

3.2. Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Two HTTP APIs have been implemented using the Flask: 1) encrypt the video; 2) verify the video.

3.3. Frontend

Figure 3 shows the basic frontend UI. It has been implemented using HTML and CSS.

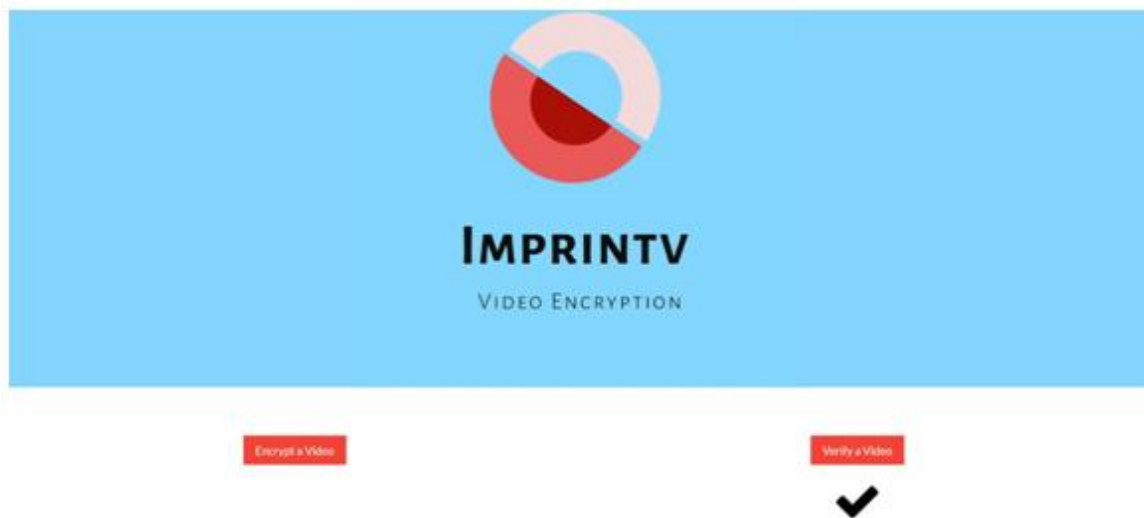


Figure 3. The Frontend Web UI

HTML - HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. Although most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

3.4. Video Encryption

Lossless compression techniques, as their name implies, involve no loss of information. If data have been losslessly compressed, the original data can be recovered exactly from the compressed data. Lossless compression is generally used for applications that cannot tolerate any difference between the original and reconstructed data.

We created a simple encryption algorithm to serve as a placeholder, but we are planning on using steganography to create a well-hidden message. The file size is too big from lossless

compression, and, the file type is limited to .avi files. We're working on utilizing lossy compression, but that's something for later.

4. EXPERIMENTS

This tool allows us to encrypt videos before release and verify them in other side to detect whether it has been edited or not. In this experiment, we applied the tool to one sample video, encrypting and verifying it (see Figure 3). We marked the frame, as well as faces every time in progress.

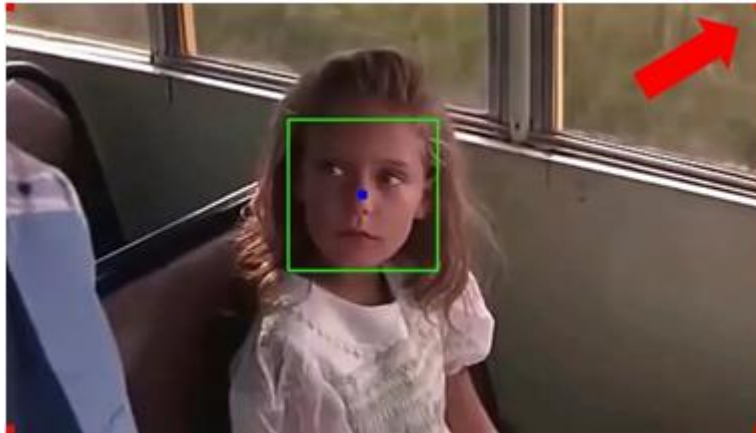


Figure 3: An experiment of encryption and decryption

5. RELATED WORK

D. Güera and E. J. Delp [1] proposed an AI-based approach to detect deepfake videos. Just like other approaches, the major disadvantage of this approach is that once the hackers understand the algorithm, they can always create a counter approach to skip the detection process. Yuezun Li [2] focuses on using an image processing technique to detect the obvious edges formed by the deepfake videos. However, as the deepfake video become more and more sophisticated, it is very difficult to guarantee the accuracy of this approach in the long run, as a number of deepfake videos cannot be judged by human eyes.

6. CONCLUSIONS AND FUTURE WORK

In this project, we proposed a video encryption approach to secure video after release. This tool uses a high efficiency video encryption algorithm to encrypt videos before release. After release, they can be verified through decryption algorithm to defect if the video has been edited or not. This tool helps us test the “authenticity” of those videos and avoid slanderous deepfakes and prevent content theft to some extent. In addition, it also secure online identity.

In the future, we will investigate other video encryption algorithms to keep improving the accuracy and efficiency. We also would like to explore the possibility of automaticity in video encryption and decryption.

REFERENCES

- [1] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.
- [2] Exposing DeepFake Videos By Detecting Face Warping Artifacts Yuezun Li, Siwei Lyu Computer Science Department University at Albany, State University of New York, USA
- [3] Ruchansky, N., Seo, S., & Liu, Y. (2017, November). Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 797-806). ACM.
- [4] Polletta, F., & Callahan, J. (2019). Deep stories, nostalgia narratives, and fake news: Storytelling in the Trump era. In Politics of meaning/meaning of politics (pp. 55-73). Palgrave Macmillan, Cham.
- [5] Singhanian, S., Fernandez, N., & Rao, S. (2017, November). 3han: A deep neural network for fake news detection. In International Conference on Neural Information Processing (pp. 572-581). Springer, Cham.
- [6] Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE.
- [7] Citron, D. K., & Chesney, R. (2018). Deep Fakes: A Looming Crisis for National Security, Democracy and Privacy?. Lawfare.
- [8] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc."
- [9] Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2012). Real-time computer vision with OpenCV. Communications of the ACM, 55(6), 61-69.
- [10] Li, Y., & Lyu, S. (2018). Exposing deepfake videos by detecting face warping artifacts. arXiv preprint arXiv:1811.00656, 2.
- [11] Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., & Verdoliva, L. (2018). Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv preprint arXiv:1812.02510.
- [12] Dolhansky, B., Howes, R., Pflaum, B., Baram, N., & Ferrer, C. C. (2019). The Deepfake Detection Challenge (DFDC) Preview Dataset. arXiv preprint arXiv:1910.08854.

TOKEN BUCKET-BASED THROUGHPUT CONSTRAINING IN CROSS-LAYER SCHEDULERS

Jeremy Van den Eynde and Chris Blondia

University of Antwerp - imec
IDLab - Department of Mathematics and Computer Science
Sint-Pietersvliet 7, 2000 Antwerp, Belgium
{jeremy.vandeneynde,chris.blondia}@uantwerpen.be

ABSTRACT

In this paper we consider upper and lower constraining users' service rates in a slotted, cross-layer scheduler context. Such schedulers often cannot guarantee these bounds, despite the usefulness in adhering to Quality of Service (QoS) requirements, aiding the admission control system or providing different levels of service to users.

We approach this problem with a low-complexity algorithm that is easily integrated in any utility function-based cross-layer scheduler. The algorithm modifies the weights of the associated Network Utility Maximization problem, rather than for example applying a token bucket to the scheduler's output or adding constraints in the physical layer.

We study the efficacy of the algorithm through simulations with various schedulers from literature and mixes of traffic. The metrics we consider show that we can bound the average service rate within about five slots, for most schedulers. Schedulers whose weight is very volatile are more difficult to constrain.

KEYWORDS

Cross-layer Scheduling, Quality of Service, Token Buckets, Resource allocation

Part of this research work was carried out at UAntwerpen, in the frame of Research Project FWO nr. G.0912.13 'Cross-layer optimization with real-time adaptive dynamic spectrum management for fourth generation broadband access networks'. The scientific responsibility is assumed by its authors.

1. INTRODUCTION

In shared communication networks users often have to compete for service. For example, in wireless networks, the total available capacity is constantly fluctuating due to interference, non-stationary objects etc. Hence, users have to cooperate with each other to ensure the available capacity is shared fairly.

Also in some wired network scenarios, such as DSL, there is competition between users due to cross-talk [1]. This cross-talk occurs in the copper cables when a user's signal leaks into other users' cables, thereby reducing the rate region, the set of all users' possible simultaneous data rates. This rate region is considered convex, implying that increasing one user's service rate, will decrease other users' rates.

In these two settings, the physical layer has many Pareto-optimal operating points, which are all considered equally good to the physical layer. To the upper layers they are, however, not all equally good: allocations that satisfy the user's QoS are preferable. For example, a live video feed has different requirements than video on demand, with regard to delay and data rates. Through cross-layering, the passing of information over the boundaries of the traditional OSI model layers, the upper layers can steer the physical layer. This cross-layering is often abstracted into a Network Utility Maximization (NUM) problem, described by Equation 1

$$\mathbf{C}^* = \arg \max_{\mathbf{C} \in \mathcal{R}} \sum_{n=1}^N u^n(C^n, \mathcal{S}) \quad (1)$$

Here n is the user index, N the number of users, and utility function $u^n(\cdot)$ is an interface between the layers, representing the value of a user receiving a service rate C^n . The state of the system \mathcal{S} can include any observable information, such as queue lengths, arrivals, delays and others. We will omit \mathcal{S} for readability when it is clear from the context. In general $u^n(\cdot)$ is an increasing, concave and differentiable function. Examples of cross-layer schedulers are mentioned in [subsection 1.2](#).

[Equation 1](#) chooses the operating point \mathbf{C}^* from the rate region (all possible combinations of service rates) such that average utility over all users is maximized. In this system, a user n 's service rate depends on the weight in relation to all other users. This implies that it is challenging to control a single user's service rate, since there is no correspondence between one user's isolated weight and its received rate. In spite of that, sometimes we want to be able to constrain the short-term (and long-term) average rates, for example to ensure the QoS of the users. We give more reasons for implementing rate constraints in [subsection 1.1](#).

Most cross-layer schedulers, such as the ones listed in [subsection 1.2](#), do not offer the option to confine the service rate. Therefore, after reviewing the system model in [section 2](#), we describe our contribution, the Token Bucket Rate Modifier (TBRM) algorithm, in [section 3](#). It is a generic low-complexity algorithm that constrains the short- and long-term average service rates of all users in a utility function-based cross-layer scheduler. Each time slot t , the NUM problem is solved, but each user's weight is replaced by $\phi^n \exp(\frac{k_g^n}{\sigma_g^n} + \frac{k_M^n}{\sigma_M^n})$. Two counters, k_g^n and k_M^n , track the deficiency and excess in service, respectively. If a user n has received less service than ρ_g^n , the amount of tokens will accumulate, and the user's weight will increase, therefore raising the probability of receiving more data rate. The parameters σ_g^n and σ_M^n are a measure for the slowness to react to deficiency and excess respectively. The variable ϕ^n accounts for non-positive weights. The algorithm is very easy to incorporate into any scheduler, as it does not require manipulating the original scheduler's weight function.

In [section 4](#), we evaluate our algorithm through multiple simulations. We compare our results with the unbounded scenarios, and look at the influence of the slot size τ and parameter σ . The metrics indicate that we can bound the average service rate for all users within a limited amount of time. Schedulers whose weight fluctuate heavily are more difficult to constrain.

We close the paper with related works in [section 5](#), and a conclusion in [section 6](#).

1.1. MOTIVATION FOR SERVICE RATE CONSTRAINTS

In a multi-user environment, it is useful to ensure that flows are guaranteed a bound on the average service rate. A provider might offer different QoS guarantees to different users, depending on the subscribed model. This might include a guaranteed and/or maximal rate.

There are other reasons to ensure a minimal rate. For example applications like audio and video need a minimal rate for a satisfying Quality of Experience (QoE). Additionally, the authors of [\[2\]](#) observe that TCP-based applications can lead to large queues when the throughput is too small. Finally, a guaranteed rate ensures that misbehaving competing flows, cannot smother flows from receiving their fair share.

Applying an upper bound on a user's capacity is also useful. For example, to accommodate a new flow into a network, admission control algorithms often require an upper bound on the data rates [\[3\]](#). If a flow disrespects this rate, other applications in the network can suffer deteriorated QoS. By limiting the maximal data rate, a misbehaving flow is isolated and cannot negatively impact the other applications, but will only punish itself. In addition, it can also be useful to provide different service levels to users, where an operator may choose to cap the data rate for cheap data services, and remove this limit for the more expensive premium services.

Although rate constraints can be implemented into the physical layer, it might be interesting to handle it at higher layers. First, it reduces the degrees of cross-layer freedom, and limits the communication necessary. Second, this allows a more flexible approach, allowing for temporary violations. Finally, it might not always be possible to introduce the rate constraints into the physical layer. For example, the scheduler is implemented in hardware or closed-source and cannot be modified, or the corresponding NUM problem's complexity might increase too much due to the additional constraints.

Imposing upper bound constraints can be easily accomplished using a token bucket counter on a user's output stream. This is wasteful though: as the medium is shared, increasing one user's service rate means decreasing other users rates. By applying a token bucket, the excess capacity is thrown away as it is reserved by a particular user.

1.2. CROSS-LAYER SCHEDULERS

There are many existing cross-layer schedulers, each employing different metrics. We list here schedulers that are used in the simulations in [section 4](#).

Most schedulers found in literature are linear, meaning that the utility is of the form $u(C^n) = \omega^n C^n$. For example, the Max-Weight (MW) scheduler, presented in the seminal work [\[4\]](#), has $\omega^n = Q^n$.

The Modified Largest Weighted Delay First (M-LWDF) [5] uses $\omega^n = \alpha^n \Gamma^n \frac{1}{C^n}$, where $\alpha^n = \frac{-\log(\epsilon)}{\hat{T}^n}$, ϵ is maximal delay violation probability, \hat{T} the delay upper bound, Γ^n is the Head-of-line delay and \bar{C}^n the exponentially averaged assigned data rates.

The Exponential/PF (EXPPF) [6] scheduler, is a combination of Proportionally Fair scheduler and an exponent. It calculates the weights for real-time flows as $\exp(\frac{\alpha^n \Gamma^n - \chi}{1 + \sqrt{\chi}}) \frac{1}{C^n}$, where $\chi = \frac{1}{N} \sum_{n=1}^N \alpha^n \Gamma^n$.

The Maximal Delay Utility (MDU) scheduler [7] employs the average waiting time: $\omega^n = \frac{|u'^n(\bar{U}^n)|}{\bar{\lambda}^n}$, where u'^n is the derivative of the traffic class' utility function, \bar{U}^n the average waiting time and $\bar{\lambda}^n$ the average arrival rate.

The MD scheduler is the non-linear equivalent of the MW scheduler, as it uses the utility function $u(C^n) = Q^n/C^n$. Finally, the Minimal Delay Violation (MDV) scheduler [8] is a non-linear scheduler of the form $u(C^n) = -\frac{\omega^n}{C^n}$, where the weight ω^n tries to minimize the amount of delay violations by looking at the queue and past delays.

2. SYSTEM MODEL

Time in our model is divided into slots of τ seconds. There are N users, indexed by $n \in [1, N]$, each of which can send $\tau \cdot C^n(t)$ bits during slot $t \in \mathbb{N}$, where $0 \leq C^n(t) \leq \hat{C}^n$ is the capacity for user n .

The user's arrivals and departures during slot t are denoted by $A^n(t)$ and $D^n(t)$ respectively while the queue at the start of slot t is indicated by $Q^n(t)$. The capacities $C(t)$ are determined by a scheduler, based on weights $\omega(t)$: at the start of slot t , a request is made to the scheduler, the reply of which is applied at the start of slot $t + 1$. There is thus a delay of τ seconds between a request and application of the rates. The scheduler takes the best matching rate from the rate region $\mathcal{R} \subset \mathbb{R}_+^N$.

Each of the N users has one traffic stream with delay upper bound \hat{T}^n , with the additional constraint that flow n 's average service rate is bounded: $0 \leq \rho_g^n \leq C^n \leq \rho_M^n \leq \hat{C}^n$.

3. THE TOKEN BUCKET RATE MODIFIER ALGORITHM

3.1. TOKEN BUCKETS

Token buckets are found in various situations, such as describing traffic flows [9-11], to check conformance of incoming or outgoing traffic (policing and shaping [12]), traffic marking in DiffServ [13-14] and rate estimation [15].

Conceptually, a token bucket $TB(\rho, \sigma)$ consists of a bucket holding k tokens (e.g. bits). Tokens are added at a constant rate ρ to the bucket, which is capped at σ tokens. Whenever a packet of B bit passes and there are sufficient tokens, B tokens are removed from the bucket and the packet continues its journey. If $k < B$, the packet is considered non-conforming and an appropriate action is taken, such as being color-marked non-conforming, queued (shaping) or dropped (policing). Such a token bucket will limit the long-term average outgoing rate to ρ . On a short-term scale, bursts of up to σ bits can be served.

3.2. ALGORITHM

In the following algorithm, this token bucket principle is used to lower and upper bound the service rate in a cross-layer scheduler setting. But, in contrast to a regular token bucket, we now do not cap the tokens to σ . Rather, they are used to indicate the severity of the excess.

In the algorithm, instead of solving

$$\arg \max_{C \in \mathcal{R}} \sum_n f(C^n) \omega^n \quad (2)$$

where $f(C) = C$ for the linear, and $f(C) = C^{-1}$ for the reciprocal variant, the NUM problem is modified to

$$\arg \max_{C \in \mathcal{R}} \sum_n f(C^n) \phi^n \exp\left(\frac{k_g^n}{\sigma_g^n} + \frac{k_M^n}{\sigma_M^n}\right) \quad (3)$$

Here, $k_g^n \in [0, \infty[$ and $k_M^n \in]-\infty, 0]$ are the tokens for the guaranteed and maximal token buckets, respectively. Every slot, the tokens are updated according to the following rules:

$$k_g^n(t+1) = \max\{0, k_g^n(t) + (\rho_g^n - C^n(t))\tau\} \quad (4)$$

$$k_M^n(t+1) = \min\{0, k_M^n(t) + (\rho_M^n - C^n(t))\tau\} \quad (5)$$

When the received capacity for a user n in the past slots is less than the guaranteed rate ρ_g^n , the virtual token counter k_g^n will continue to increase as long as there is a deficit in received service, and hence the weight will exponentially increase. Likewise, if a user n has received more than ρ_M^n service, the virtual token counter k_M^n will have a negative drift, as long as more data rate is assigned to the user. This will reduce the user's weight exponentially. When the service rate is less than ρ_M^n , the token counter will return to 0.

We introduce

$$\phi^n = \begin{cases} \bar{\omega}, & \text{if } \omega^n \leq \epsilon \text{ and } \frac{k_g^n}{\sigma_g^n} + \frac{k_M^n}{\sigma_M^n} \neq 0 \\ \omega^n, & \text{else} \end{cases} \quad (6)$$

to account for non-positive weights ω^n . Here $\bar{\omega}$ is the exponentially averaged sum of all the positive weights, and ϵ a small number that will result in a capacity close to zero (for the simulations we used $\epsilon = \max_n \{\omega^n\} \cdot 10^{-5}$). If $\omega^n \in]0, \epsilon]$ it becomes difficult to increase the bandwidth reliably, and in the case of $\omega^n = 0$, it is even impossible, since the weight will remain zero. For a negative weight, which can occur for example for best effort flows in the EXP/PF scheduler, the additional factor would just result in an even lower weight, and would also inhibit us from receiving service. $\bar{\omega}$ is used to approximate a valid weight that is reasonably stable. This weight is scheduler and traffic dependent, and thus must be calculated at run time.

Note that if $\rho_g^n = 0$ or $\rho_M^n \geq \hat{C}^n$, then respectively the first and second exponent will always be 1, and the respective bound is disabled.

It can be seen that if a flow stays within the bounds, then the tokens k_g^n and k_M^n will remain zero, and $\phi^n = \omega^n$, resulting in the unmodified weight. Only if some rate guarantee will not be met, weights will be adapted.

3.3. DISCUSSION

Parameters ρ_g^n and ρ_M^n

The choice of ρ_g^n and ρ_M^n influences the speed at which the rate can adapt. For example, if the guaranteed rate $\rho_g^n = 0.75\hat{C}^n$, then in each slot the tokens can increase by at most $0.75\hat{C}^n$, and the negative drift is at most $0.25\hat{C}^n$. Thus, if this flow has been receiving no service, then the tokens - and thus the weight too - will increase quickly. If it is receiving service at a rate \hat{C}^n , then the tokens will decrease more slowly. A small ρ_g^n thus also implies a small positive and large negative drift. A similar reasoning can be applied to the maximal rate ρ_M^n .

Parameters σ_g^n and σ_M^n

In the traditional token bucket algorithms, σ is a measure for the burstiness of a flow. For example, large values of σ mean that large bursts are allowed. In our algorithm, the σ can be interpreted as a measure for slowness to react. A large value of σ_M^n means that longer periods of above-guaranteed service rates are possible, because our weight will decrease more slowly. Small values of σ_M^n will react quicker and can lead to an overreaction. The two token buckets can also influence each other: in case of an overreaction, the other token bucket will also have a sudden excess, and in turn have a fiercer reaction. This can be observed for small values of σ in the simulations of [subsection 4.5.2](#).

Slot size

In our system, the slot size implies a delay between a request for and subsequent assignment of the capacity. A larger slot size means that changes will be slower, and that predicting future traffic becomes more important. This also matters to the rate constraint algorithm, since the scheduler's response to weights becomes more unpredictable, hence modifying the weights. The simulations of [subsection 4.5.3](#) briefly look at increasing slot sizes.

exp

The function exp is chosen here to modify the token fractions, but any continuous, strictly increasing function $\alpha(\cdot)$ for which holds that $\alpha(0) = 1$, $\lim_{x \rightarrow -\infty} \alpha(x) = 0$ and $\lim_{x \rightarrow \infty} \alpha(x) = \infty$ will give rate guarantees, albeit with different bounds. Tests with different functions resulted in more short-time erratic behavior.

Additive form

Instead of using a product, it is also possible to use an additive form,

$$\arg \max_{C \in \mathcal{R}} \sum_n f(C^n) \omega^n + \left(\alpha\left(\frac{k_g^n}{\sigma_g^n}\right) + \alpha\left(\frac{k_M^n}{\sigma_M^n}\right) \right) \beta$$

Here α is a continuous, strictly increasing function with the properties $\alpha(0) = 0$, $\lim_{x \rightarrow -\infty} \alpha(x) = -\infty$ and $\lim_{x \rightarrow \infty} \alpha(x) = \infty$. An additional factor β must be introduced to account for the fact that ω^n is usually not unitless.

We ran some simulations for $\alpha(x) = x$ and $\alpha(x) = x^3$, and $\beta = \bar{\omega}$. The simulations showed that this approach is also possible, and avoids the non-positive weight problem which forced us to introduce the factor ϕ^n . However, in the NUM problem, what matters is the relative weights, rather than the absolute difference, which the additive form expresses. Even though on larger timescales this leads to nicely averaged data rates, on short timescales the behavior is very extreme, where $C^n(t)$ alternates between 0 and rates close to \hat{C}^n in successive slots.

Complexity

The space and time complexity of the TBRM algorithm is very low. Every slot we update the N users' token counters k_g^n and k_M^n (Equations (4) and (5)). Additionally, we have to select a suitable ϕ^n for all n . The exponentially weighted $\bar{\omega}$ is a constant time operation $\mathcal{O}(1)$. The resulting time complexity is thus $\mathcal{O}(3N + 1) = \mathcal{O}(N)$.

Likewise, the space requirements are equally low: we track the $2N$ counters, and a single exponentially weighted $\bar{\omega}$. The space complexity is in this case $\mathcal{O}(2N + 1) = \mathcal{O}(N)$.

Other considerations

Applying rate guarantees transforms a work-conserving scheduler into a non-work conserving scheduler. I.e. the scheduler might have capacity assigned, even though there are no jobs available.

Additionally, throughput constraints reduces the stability region of a scheduler. Roughly, a scheduler is called stable for an arrival process if all the queues remain bounded. The set of arrival rates for which a scheduler is stable, is called the stability region. Schedulers such as MW [4] and MDU [7] have been proven to be stable for the widest range of arrivals.

Applying a constraint on the data rate for such schedulers reduces its stability region. Enforcing a maximal data rate inside the stability region, clearly decreases this region. Also supporting a minimal throughput constraint influences the stability region: a minimal throughput constraint can be rewritten as a (more complex) maximal throughput constraint on the other users.

4. SIMULATIONS

4.1. SIMULATION SETUP

We evaluated the TBRM algorithm using simulations for the schedulers listed in subsection 1.2. We ran simulations in OMNeT++ using the INET framework. Every $\tau = 50\text{ms}$ the original weights and weight modifiers were computed. The resulting NUM problem was then solved with the help of the nlopt [16] library, by first applying the local variant of the DIviding RECTangles algorithm [17], followed by the COBYLA algorithm [18], to obtain the final rate, applying them in the next slot.

4.2. RATE REGION

The rate region was artificially generated by the following formula, which is based on the n-sphere formula.

$$r^n = \hat{C}^n \prod_{i=1}^{n-1} \sin(\phi^i)^{1-\gamma} \cos(\phi^n)^{1-\gamma}, r \in [1, N-1]$$

$$r^N = C_{max}^N \prod_{i=1}^{N-1} \sin(\phi^i)^{1-\gamma}$$

For N users, the rate region is the set of points generated by varying $\phi^i \in [0, \frac{\pi}{2}]$, $\forall i \in [1, N-1]$. The modifier $\gamma \in [-1, 1]$ changes the shape of the rate region. If we set $\hat{C}^n = M$, $\forall n$, then the shape ranges from an n-simplex for $\gamma = -1$, over an n-sphere with radius M for $\gamma = 0$, to a hypercube for $\gamma = 1$.

4.3. SCENARIOS

The scenarios listed in Table 1 show the different types of traffic and the applied constraints.

The traffic types behave differently on short and large timescales. The first type of traffic consists of a sine-wave, superimposed with a faster oscillating sine-wave. Some flows will oscillate slowly (Sine2VS) while others oscillate fast (Sine2F). The second type of traffic is the heavy tail traffic, which is either a trace file of a video file, such as Starwars, or a self-similar flow, generated by a superposition of Pareto-distributed sources [19]. The last class of traffic, SAT, tries to send as much traffic as possible, by ensuring the queue is always backlogged.

These scenarios are run for $\tau = 0.05\text{s}$ in subsection 4.5.1. In subsection 4.5.2, we vary σ , and in subsection 4.5.3, it is τ that changes.

Scenario	User 1	User 2	User 3	User 4	User 5
1	SAT [150,250]	SAT [250,350]	SAT [350,400]	SAT [150,350]	SAT [50,100]
2	Starwars [50,150]	Alice [250,350]	Self-Similar [150,350]	SAT [150,350]	Sine2VS [50,120]
3	Starwars [50,150]	Sine2F [250,350]	Self-Similar [150,350]	SAT [150,350]	Sine2VS [50,120]
4	Sine2VS [150,250]	Sine2VS [150,250]	Sine2VS [250,300]	Sine2VS [150,350]	Sine2VS [50,400]
5	Sine2VS [150,250]	Sine2VS [150,250]	Sine2VS [250,300]	Sine2VS [150,350]	Self-Similar [0,0]

Table 1: Summary of scenarios. Listed for each user are traffic type, and $[\rho_g, \rho_M]$ in Mbps.

4.4. METRICS

We examined three different metrics. The m2 and m3 metrics are defined on windows of size G , which groups G consecutive slots.

- m1 the percentage of slots that would be marked non-conforming by a token bucket process $TB(\rho_g, \rho_g \tau x)$ and $TB(\rho_M, \rho_M \tau x)$ for respectively the guaranteed and maximal rate. $x \in \mathbb{R}^+$ is a variable indicating the allowed burstiness. Increasing x allows for more burstiness, and will result in a smaller percentage of non-conforming slots.
- m2 The average amount of excess bits per window G . If we define the amount of bit reserved in window w as $C^G(w) = \sum_{t=w}^{(w+1)G} C(t)\tau$, and W as the total number of windows, then the m2 metric for respectively the guaranteed and maximal rate can be formally described as $\langle \{\max\{\rho_g G \tau - C^G(w), 0\} | w = 0..W - 1\} \rangle$ and $\langle \{\max\{C^G(w) - \rho_M G \tau, 0\} | w = 0..W - 1\} \rangle$. This is a representation of the severity of the average violation. A larger number indicates more severe violations. The metric can be visualized by imagining the surface above or below the required rate. Increasing G decreases the m2 metric as we smooth out excess bits over a larger window.
- m3 $\langle B^G \rangle$: where B^G is the set of consecutive violating windows. This metric gives an idea of how grouped violations are. For example, if this number is large, it means that a violation is resolved slowly.

4.5. RESULTS

4.5.1. REGULAR SCENARIOS

In the following plots, we averaged over all schedulers and scenarios, as showing the individual schedulers would result in a cluttered plot. Important discrepancies between schedulers will be discussed in the text.

Each plot has two curves, one of which displays the results for which no rate constraints were applied, as a base case, and the other has our TBRM algorithm applied.

m1

The m1 metric is shown in [Figure 1](#), which displays on the x-axis the allowed burstiness, and on the y-axis the percentage of non-conforming slots.

If we examine [Figure 1a](#), which shows the m1 metric for the upper bound, then we can see that for $x = 1$, the number of violations is close to the results of the unconstrained simulations. Increasing x , the allowed burstiness, however, we can observe that the violation probability quickly drops for our TBRM algorithm, and becomes almost 0 when the allowed burst size is $5\rho_M \tau$. This indicates that the violations occur irregularly spread. The unconstrained results remain fixed around 6% for a long time.

The underlying data shows that for the constrained scenarios all the schedulers inhibit the same behavior: there is a steep decline in violations, going from $x = 1$ to $x = 2$, and then they gradually go to almost 0 for $x = 5$.

This behavior is the same for all schedulers over all traffic classes. However, the initial violation probability for SAT class is slightly lower than the video, self-similar and sine classes. The SAT traffic is easier to correct due to its queue based nature.

In the m1 plot of the guaranteed rate in [Figure 1b](#), our domain is limited to $]0, 1]$: if $x = 1$, then it means that approximately in every slot we allow a deficit of $\rho_g \tau$ bit, which is obviously the maximum deficit we can attain per slot. In the plot, one can see that there is a much wider gap between the constrained and unconstrained scenarios, confirming the efficacy of our algorithm.

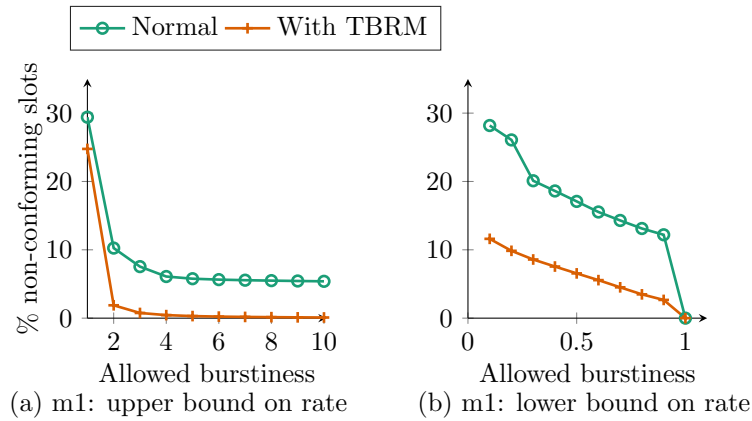


Figure 1: m1 for the regular scenarios

The data shows here that the majority of the violations come from the MW and M-LWDF schedulers, and more specifically for the video streams. For example, in the TBRM scenarios, for $x = 0.1$, both schedulers have a violation probability of about 20%, while the other schedulers are closer to 6%.

This difference can be explained by the fact that in those linear schedulers the queue length is used as a weight. This number is immediate, which causes a more unpredictable weight (especially in combination with a linear scheduler), making it more difficult to estimate a suitable weight modifier. Additionally, the weight can become 0 very easily. This requires the use of the additional ϕ^n modifier. Even though $\bar{\omega}$ is smoother, the switch between $\bar{\omega}$ and ω^n can also be disruptive. However, this extra factor is necessary, as simulations without this correction ϕ^n , result in a much higher violation probability.

The curve looks quite linear. This can be explained by the fact that the guaranteed rate violations are more evenly spread out.

m2

Ideally, we can limit the rate immediately. However, there is an inherent delay of 1 slot, and an elasticity in the form of a burst factor. Therefore, we study the rate, when we group $\frac{G}{\tau}$ slots into windows of size G .

The m2 metric in [Figure 2](#) displays the average amount of violated bits per window, for increasing window sizes G .

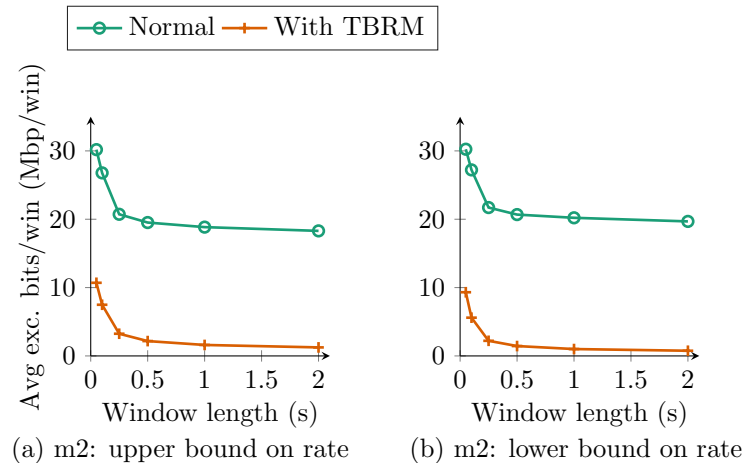


Figure 2: m2 for the regular scenarios

It can be observed that for a window size of $G = 0.05s$, for the unconstrained scenarios there is an average of 30Mbit/window in excess of the target rate. When we constrain it using our algorithm, this drops to about 10Mbit/window.

Increasing the window size G , averages out bursts. Like the results for m1, there is a steep decline until $G = 0.25s$, which coincides with 5 slots, after which the bit violations remains stable in the upper bound case. Though less pronounced, also here do the MW, M-LWDF and MD schedulers fare the worst for small window sizes, mainly for the Self-Similar traffic.

The decline implies that bursts are usually short-lived: overflow and good windows are usually close together, as they don't violate the constraints when merged. This is also confirmed in the m3 metric, below. The rate of decline is similar for the scenarios with and without TBRM applied.

m3

The last metric discusses the average length of a violation streak. [Figure 3](#) shows the average number of successive windows that violate their constraints in a log-plot. The m3 metric, like the m2 metric, initially decreases quickly as the window size increases, and then slowly decreases. It can be clearly seen that, regardless of the unconstrained behavior, the TBRM algorithm limits the bursts to 5 windows, for $G = \tau$, dropping to 2 windows for $G = 5\tau$. These short bursts confirm that the algorithm is able to fix excesses within about 5 slots.

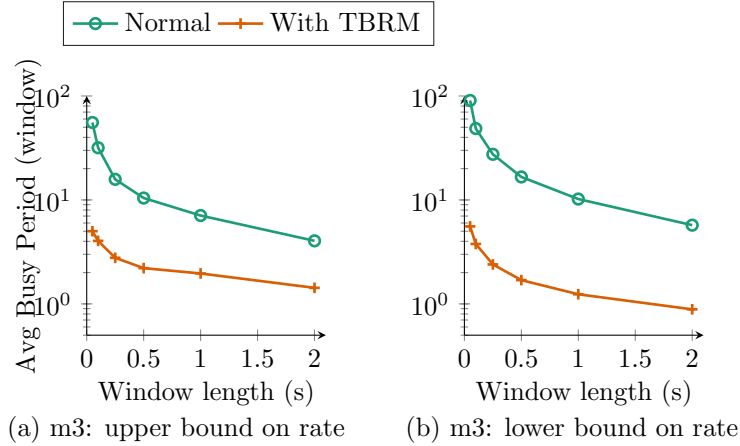


Figure 3: m3 for the regular scenarios

The main contributor to the average in this metric is the MDU scheduler. The data show that this is because whereas other schedulers consist of many smaller busy periods, the MDU scheduler has only one or two large busy periods, increasing the average significantly.

Without the MDU data, the average for 5 windows is about 1, for the minimal rate constraint, and 2 for the maximal rate constraint.

4.5.2. STUDY OF PARAMETER σ

In this section we look at the results of varying burst parameter σ . We let $\sigma_g = i\tau\rho_g$ and $\sigma_M = i\tau\rho_M$, for $i \in [10^{-2}, 10^4]$, and look at the effect on the m1 metric in [Figure 4](#), which shows the results for the individual schedulers.

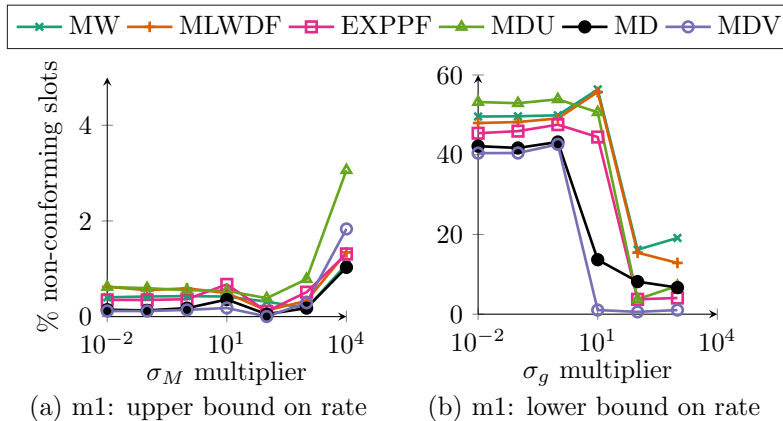


Figure 4: m1 for the σ scenarios

The plot shows that for the upper bound constraints, the violation probability is always very low (about 4% at most for $i = 10^4$). The lower bound, however, starts around 50% violation probability, and suddenly drops to smaller probabilities for $i = 10^2$.

Indeed, a small σ_M will overflow quickly, which causes its weight to be reduced swiftly, hence there will be less violations. As σ_M grows, the weight modifier will decrease much more slowly, leaving more room for violations. For the guaranteed rate, on the other hand, k_g cannot build up a deficit as fast as k_M , as discussed before. It is only when the growths of the deficits are balanced that the m1 metric can lower, which is around $i = 10^2$ and upwards.

The schedulers that perform the worst are, unsurprisingly, the MW and M-LWDF schedulers.

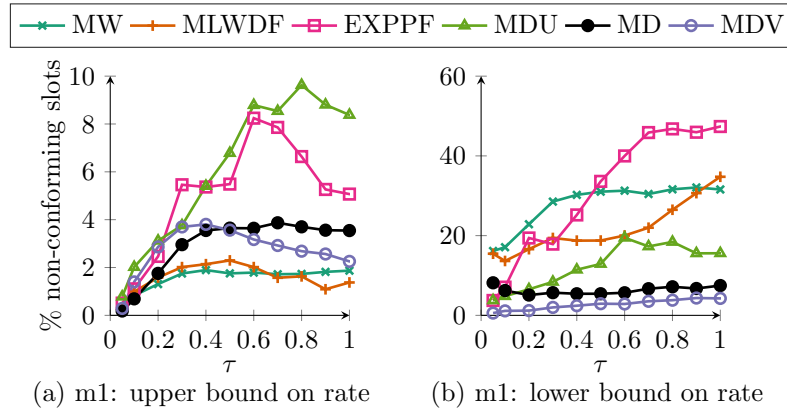


Figure 5: m1 for the τ scenarios ($\sigma = 5\tau\rho$)

4.5.3. STUDY OF PARAMETER τ

In the previous simulations, we assumed a slot length of $\tau = 0.05s$. This study observes how the TBRM algorithm changes in function of τ .

In [Figure 5](#) the m1 metric for $\sigma = 5\tau\rho$ is plotted. It can be observed that mainly the lower bound in [Figure 5b](#) is sensitive to an increasing slot length. This might be because with an increasing τ also grows the probability of a larger delay: if in a slot a low capacity was assigned erroneously, the delays or queues will increase and additionally it takes longer to correct, causing larger queues. Especially the EXPPF scheduler suffers from this, as it is of the form $\exp(\Gamma)$, where Γ is the Head-of-line. As the non-linear Min-Delay (MD) and MDV schedulers try to minimize the delay, they suffer less from an increase of slot size.

For the upper bound in [Figure 5a](#) only the MDU and EXPPF schedulers seem to suffer from the increased slot size. This is probably due to the fact that it is easier to receive a service lower than ρ_M .

5. RELATED WORK

In [\[5, 20\]](#) the authors use virtual tokens as a measure for the average waiting time, and incorporate it with the M-LWDF [\[5\]](#) and EXPPF [\[6\]](#) scheduling algorithm to warrant a minimal rate. It is, however, not transferable to other schedulers. In other schedulers, the guaranteed rate constraint is built into the scheduler itself [\[21, 22\]](#), but they are all scheduler-specific and don't allow enforcing a maximal data rate. The authors of [\[23\]](#) consider utility based throughput allocation subject to certain properties, but is only valid for linear utility functions. In [\[24\]](#) a related problem of maintaining an optimal service rate is proposed. In [\[25\]](#), the authors consider a generic algorithm with minimum and maximum rate constraints. It is, however, only applicable to schedulers that operate in function of an average rate. As such, it excludes for example the MW [\[4\]](#), MD and M-LWDF schedulers. Other schedulers, such as MDU [\[7\]](#) and MDV [\[8\]](#) have a more elaborate utility function and are more difficult to characterize. The authors of [\[26\]](#) also employ a token system, but assume that users lie about their demands to strategically maximize their utility. In [\[27\]](#) constraints are applied to network slices of traffic aggregates in a 5G context, using an additive approach.

6. CONCLUSION

In this paper we looked at restricting the service rates given to users in a cross-layer scheduler setting. We implemented this using a low-complexity algorithm that modifies the weights in a Network Utility Maximization problem, using the concept of token buckets. We first discussed cross-layer scheduling, and the need to both upper and lower limit data rates assigned to users. Then we proposed the TBRM algorithm, and followed up with simulation results. We ran simulations for six different schedulers, and

multiple scenarios demonstrating that using our approach it is possible to limit the service rate, within error, after about five slots for the maximal and guaranteed service rate for most schedulers. Schedulers that progress smoothly are easier to constrain than schedulers that can behave wildly, such as EXPPF for long slot times, MW or M-LWDF. These are more difficult to restraint with respect to guaranteeing a lower bound on the service rate.

REFERENCES

- [1] J. Verdyck, C. Blondia, and M. Moonen, "Network utility maximization for adaptive resource allocation in dsl systems," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 787–791.
- [2] R. Chakravorty, S. Katti, J. Crowcroft, and I. Pratt, "Flow aggregation for enhanced tcp over wide-area wireless," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE Societies, vol. 3. IEEE, 2003, pp. 1754–1764.
- [3] J. Qiu and E. W. Knightly, "Measurement-based admission control with aggregate traffic envelopes," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 2, pp. 199–210, 2001.
- [4] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE transactions on automatic control*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [5] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications magazine*, vol. 39, no. 2, pp. 150–154, 2001.
- [6] R. Basukala, H. M. Ramli, and K. Sandrasegaran, "Performance analysis of exp/pf and m-lwdf in downlink 3gpp lte system," in *Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on*. IEEE, 2009, pp. 1–5.
- [7] G. Song, "Cross-layer resource allocation and scheduling in wireless multicarrier networks," Ph.D. dissertation, Citeseer, 2005.
- [8] J. Van den Eynde, J. Verdyck, M. Moonen, and C. Blondia, "A delay-based cross-layer scheduler for adaptive dsl," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.
- [9] P. P. Tang and T.-Y. Tai, "Network traffic characterization using token bucket model," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. IEEE, vol. 1. IEEE, 1999, pp. 51–62.
- [10] R. L. Cruz, "A calculus for network delay. i. network elements in isolation," *IEEE Transactions on information theory*, vol. 37, no. 1, pp. 114–131, 1991.
- [11] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queueing systems for the internet*. Springer Science & Business Media, 2001, vol. 2050.
- [12] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking (ToN)*, vol. 6, no. 5, pp. 611–624, 1998.
- [13] J. Heinanen and R. Guerin, "Rfc 2697—a single rate three color marker," *IETF, September*, 1999.
- [14] —, "Ietf rfc 2698." *A Single Rate Three Colour Marker*, 1999.
- [15] E. Zhang and L. Xu, "Capacity and token rate estimation for networks with token bucket shapers," *Computer Networks*, vol. 88, pp. 1–11, 2015.
- [16] Steven G. Johnson, "The nlopt nonlinear-optimization package." [Online]. Available: <https://github.com/stevengj/nlopt>
- [17] J. M. Gablonsky and C. T. Kelley, "A locally-biased form of the direct algorithm," *Journal of Global Optimization*, vol. 21, no. 1, pp. 27–37, 2001.
- [18] M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in optimization and numerical analysis*. Springer, 1994, pp. 51–67.
- [19] X. Bai and A. Shami, "Modeling self-similar traffic for network simulation," *arXiv preprint arXiv:1308.3842*, 2013.
- [20] S. Shakkottai and A. L. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in hdr," in *Teletraffic Science and Engineering*. Elsevier, 2001, vol. 4, pp. 793–804.
- [21] M. Mohseni, R. Zhang, and J. M. Cioffi, "Optimized transmission for fading multiple-access and broadcast channels with multiple antennas," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1627–1639, 2006.
- [22] X. Wang and N. Gao, "Stochastic resource allocation over fading multiple access and broadcast channels," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2382–2391, 2010.
- [23] X. Liu, E. K. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer networks*, vol. 41, no. 4, pp. 451–474, 2003.
- [24] S. Borst and P. Whiting, "Dynamic channel-sensitive scheduling algorithms for wireless data throughput optimization," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 3, pp. 569–586, 2003.
- [25] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4. IEEE, 2005, p. 2415–2424.

- [26] S. M. Zahedi, S. Fan, and B. C. Lee, "Managing heterogeneous datacenters with tokens," *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 2, p. 18, 2018.
- [27] S. Mandelli, M. Andrews, S. C. Borst, and S. Klein, "Satisfying network slicing constraints via 5g mac scheduling," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2332–2340, 2019.

MEASUREMENT AND CHARACTERIZATION OF THE STATIONARY NOISE IN NARROWBAND POWER LINE COMMUNICATION

Raja Alaya and Rabah Attia

Tunisian Polytechnic School, University of Carthage, Tunisia

ABSTRACT

Understanding the interference scenario in power lines network is a key step to characterize the power line communication (PLC) system. This paper focuses on the characterization and modelling of the stationary noise in Narrowband PLC. Measurement and analysis of noise is carried out in the Tunisian outdoor Low Voltage (LV) power line network in the frequency band below 500 kHz. Based on existing models and measurements results, a parametric model of noise is proposed; the model parameters are statistically studied.

KEYWORDS

Power Line Communication, Measurement, Modelling, Narrowband Frequency, Noise;

1. INTRODUCTION

In the last few years, narrowband (i.e., frequency range between 9 and 500 kHz) power line communication (NB-PLC) technology has attracted the industry and tends to be a promising way of information exchange. This technology is widely used in Smart Grid (SG) applications as a communication infrastructure for advanced metering infrastructures communications, automatic meter reading and demand response [1]. The main advantage of PLC is to use the electrical network for the communication which is an existing infrastructure and it potentially covers the entire country under study. This permits to avoid additional wiring and have a quick deployment. However the existing power lines were originally not designed for signal transmission, but only for electricity delivery for end customer. Otherwise LV power lines present a very harsh communications media which suffers from several kinds of disturbances, such as the stationary noise and the impulsive noise. The presence of the noise in PLC system is a crucial factor that affects the transmitted signals. In fact, the performance of a communication system is directly related to the noise level [2]. To this extent, it is useful to have as much knowledge as possible of the noise by identifying the sources of interference and developing a mathematical model which adequately describes noise characteristics in power line communication networks.

To characterize noise and develop its models, statistical knowledge of noise parameters are required that can acutely describe noise present in power line, which can be obtained from experimental measurement. Therefore a lot of measurement campaigns should be done.

A lot of works have already characterized the impulsive noise and stationary noise. However they are primarily proposed in the case of Indoor at the Broadband frequency [3, 4, 5]. The

available noise measurements carried out in LV distribution networks and Narrowband frequency are mostly conducted to only describe the impulsive noise, and there has not been much attention to the stationary noise. Even if many studies are available in the literature [6, 7, 8], there is no works have been performed on the Tunisia distribution LV network. In this context, the manuscript proposes the following two main contributions: the first one is measurement and analysis of the stationary noise present in the LV distribution networks in Tunisia. The second contribution is to propose a statistical parametric model based on the analysis of the measured noise. The parameters of the model are considered as random variables approximated by their corresponding statistical distributions. This model can be used for advanced studies as a noise generator.

In this manuscript, the state of art is described in Section 2. In particular, the sources of interferences in PLC and the existing models of noise are presented. Section 3 explains the experimental measurements' setup and the main results: the spectral analysis of the NB-PLC noise in the various locations. Section 4 presents characterization and modelling of the noise in the frequency domain. Finally, the paper is concluded in Section 5.

2. STATE OF ART

2.1. Sources of Interferences

Unlike the other wired transmission structures, noise present in power-line cannot be described with an Additive White Gaussian Noise (AWGN), and it is hard to characterize it with one universal model. In fact, the PLC channel is as a hostile environment for data transmission and the noise scenario is rather complicated and exhibit quite different behaviours due to the presence of different sources of disturbance. As can be seen in Figure 1, power line noise can be separated in the following classes according to their spectral and time behaviours [9], [10]:

- 1) Colored background noise: it is always present in the network. It represents the summation of the noise generated by different noise sources. Its Power Spectral Density (PSD) increases towards lower frequencies.
- 2) Narrow-band noise: it is a radio frequency interference primary originates from the broadcasting stations. The amplitude can be changed in dependence on time and place.
- 3) Asynchronous periodic impulsive noise. This kind of noise can be considered as cyclostationary and synchronized with the mains; it is mostly caused by switched mode power supplies.
- 4) Synchronous periodic impulsive noise. It is a cyclostationary noise that synchronous with the mains. It is mainly caused by switching actions of rectifier diodes which occurs synchronously with the mains cycle.
- 5) Asynchronous impulsive noise. It is mainly generated by switching transients in the network; it is the most harmful noise source.

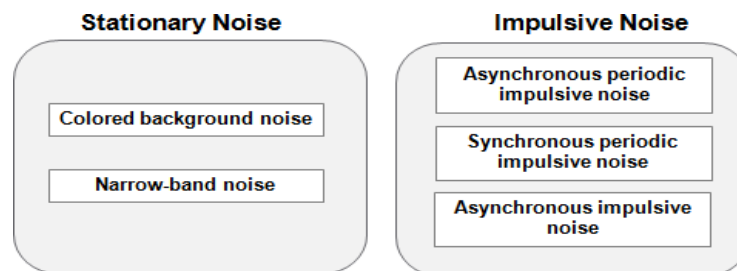


Figure 1. Classification of Noise on PLC Channels.

According to the literature [11], the amplitudes of the two first types vary slowly over time from seconds to even hours, so that they can be summarized as stationary noise. Then again, the amplitudes of the last three types change rapidly; they are defined as a set of single pulses or bursts with short durations from microseconds to milliseconds. Therefore, they can be regarded as impulsive noise. In this paper, our interest is mainly focused in types (1) and (2), while types (3), (4) and (5) will be studied in future works.

2.2. Noise Models

Noise models available in the technical literature are obtained based on empirical measurements in the frequency or time domain. Generally, background noise is modelled in the frequency domain [12], while the impulsive noise is modelled in both frequency and time domains [13]. The well existing techniques to model the noise are developed by fitting the measured noise PSD into certain functions of frequency. Moreover, noise models are obtained based on fitting the statistical properties of power line noise. Many researchers have already studied noise of type (1) and type (2) and many noise models are available in the literature, the most famous and well accepted PLC noise models for the colored background noise are the Esmalian model [14] and the OMEGA model [15]. The OMEGA model is a frequency domain model for the colored background noise. This model, based on PSD fitting noise, is characterized by a logarithmic decay function. The Esmalian model is also a frequency domain model for the colored background noise. It is based on measurements and depicts the background noise as a power frequency dependent function. Another model proposed by Philipps [16], suppose that the noise PSD is a first order exponential function, where the noise PSD decreases as the frequency gets larger. For the Narrowband noise, the most common model is a Gaussian function represented by summation of various signals. In [17] the Narrowband noise is modelled as the sum of multiple sine signals with different amplitudes at specific frequencies. The stationary noise model is basically regarded as the superposition of background noise model and narrowband noise model [18].

3. NOISE MEASUREMENT SETUP AND RESULTS

3.1. Experimental Measurement Setup

The measurement setup of the frequency domain noise is performed with a spectrum analyzer GSP-830 from GW INSTEK, configured with a resolution bandwidth of 3 kHz, and a sweeping time of 80 ms on the total band up to 500 kHz. It is important to note that all measurements are carried out in the same configuration. As shown in Figure 2, the spectrum analyzer is connected to the power line through a coupling unit which is designed by the authors in [19]. The coupling unit, as the name implies couple and decouples the high frequency signal from or to the power line. In other words, it blocks the 50/60 Hz current from entering the measurement instrument, it also prevents the high voltage of the mains from damaging the measurement instrument.

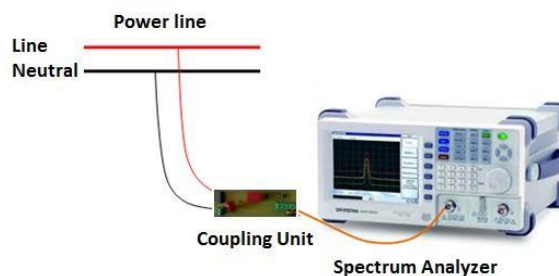


Figure 2. Experimental setup to measure the stationary noise

The schematic diagram of the coupling unit is shown in Figure 3.

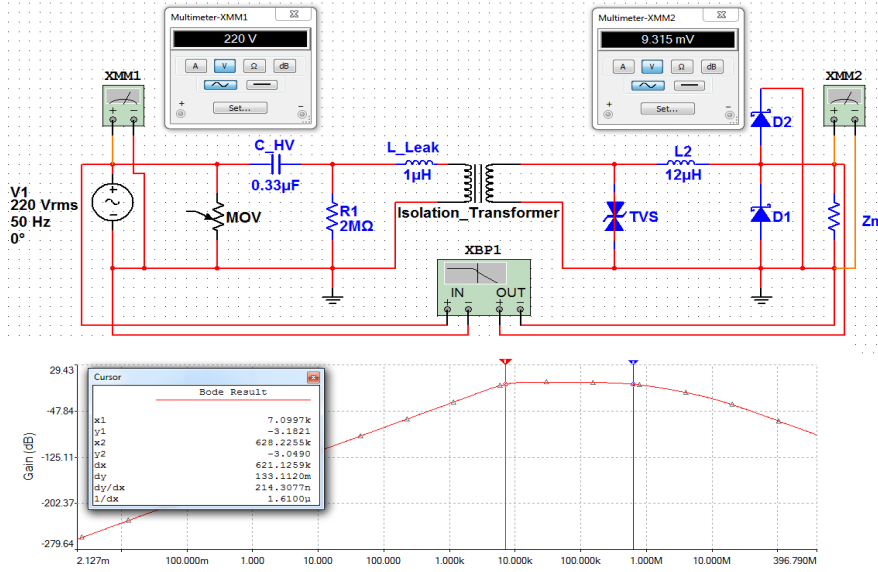


Figure 3. Designed Coupling Unit and Simulated band-pass filter.

During two weeks, many measurements were performed at diverse times of the day, in three typical urban sites in Tunisia: the first one, designed in the following site (S1), is an underground network; this site is modern and simple. The second one is dense, designed in the following site (S2), it is a mix of both underground and overhead lines; the customers in this site can be commercial or residential. The third one, designed in the following site (S3), is extremely dense and complex; it is an overhead network in which all the customers are residential.

3.2. Experimental Results and Discussion

Figure 4 and Figure 5 illustrates respectively the evolution of the noise PSD in the three typical sites in the customer side and the evolution of the noise PSD in the transformer substation.

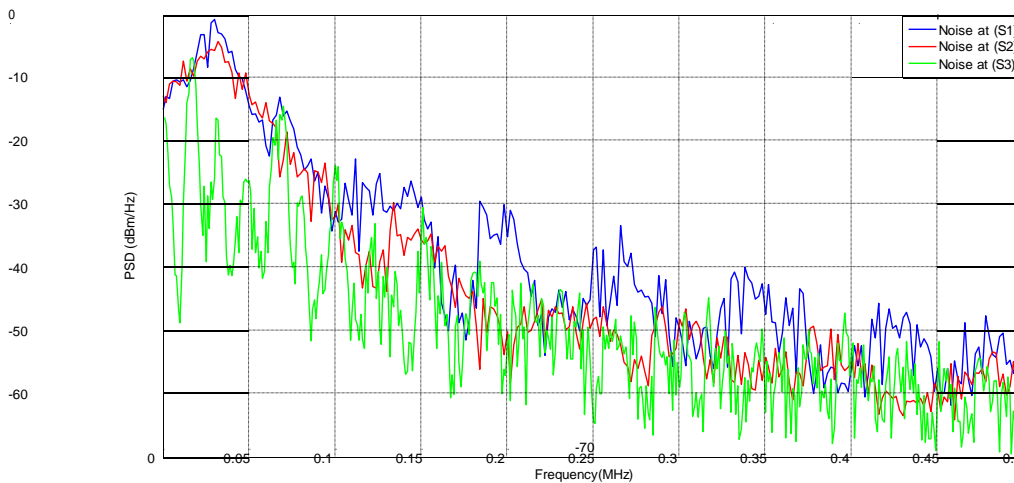


Figure 4. Noise PSD at the customer side at the meter in the three typical sites.

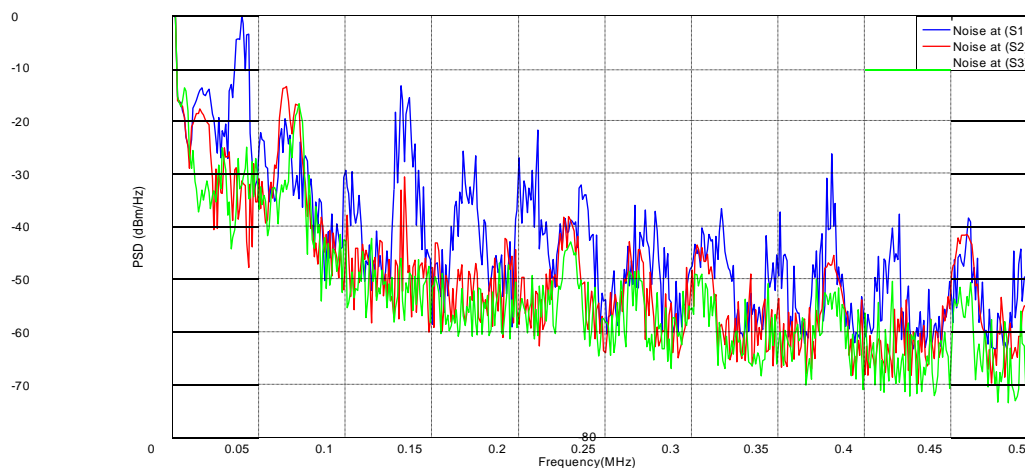


Figure 5. Noise power spectral density (PSD) in the transformer substation in the three typical sites.

It is noted that the noise has a frequency dependent nature. The noise level is higher at the lower frequencies and the level decreases as the frequency increases. The shape of the noise is a decreasing function of frequency with the presence of narrowband interferences. The interferences appear in different frequencies with random level and bandwidth. This is caused by the presence of many sources of low-frequency noise in the power network.

Comparing the noise level in the three sites, we can say that the noise level measured at the site (S3) is significantly higher than the noise level measured at the site (S1) and the site (S2), this may be explained by the important number of loads connected at the same time and topology of the network which is very complex and extremely dense. Comparing the noise measured at the site (S1) and at the site (S2), we can say that the number of interferences increases at the site (S2), because the usage of electrical appliances is significant, since there is a commercial consumer in this site. However, it can be noted that the shapes of the measured noises are in general similar.

Finally, in comparison with the customer side, the number and the depth of the interferences at the transformer side are high, because the number of loads (customers) connected to the transformer substation is more important. Hence, noise measured in the transformer substation is the sum of noise generated by all domestic loads connected to the same branch.

4. NOISE CHARACTERIZATION AND MODELLING

The stationary noise in the PLC environment is regarded as the superposition of the background noise and the narrowband interferences, which is the basis of the following modelling.

4.1. Colored Background Noise

Based on empirical measurements, the noise model is obtained by fitting the measured noise PSD into certain functions of frequency. In the literature, several models of colored background noise are proposed [14, 15].

Since, the background noise PSD is a decreasing function of frequency, the author use the Esmalian model, where the noise is considered Gaussian and it is described as follow:

$$N_{CB} = af^b + c \quad (1)$$

Where N_{CB} stands for the Colored Background noise PSD in dBm/Hz, f denotes the frequency in Hz, a , b and c are the model parameters derived from measurements. The parameter a controls the noise floor, b controls the value of noise PSD at starting frequency and c controls the form of the frequency dependent decay.

Figure 6 illustrates the corresponding fitting (bold curve) of the measured background noise in two cases.

It is observed that for stronger noise, we have the worst case with parameters values: $a = -66.76$, $b = 0.3942$, $c = -12.9$. For weaker noise, we have the best case with values: $a = -76.95$, $b = 0.25$, $c = 0.03$.

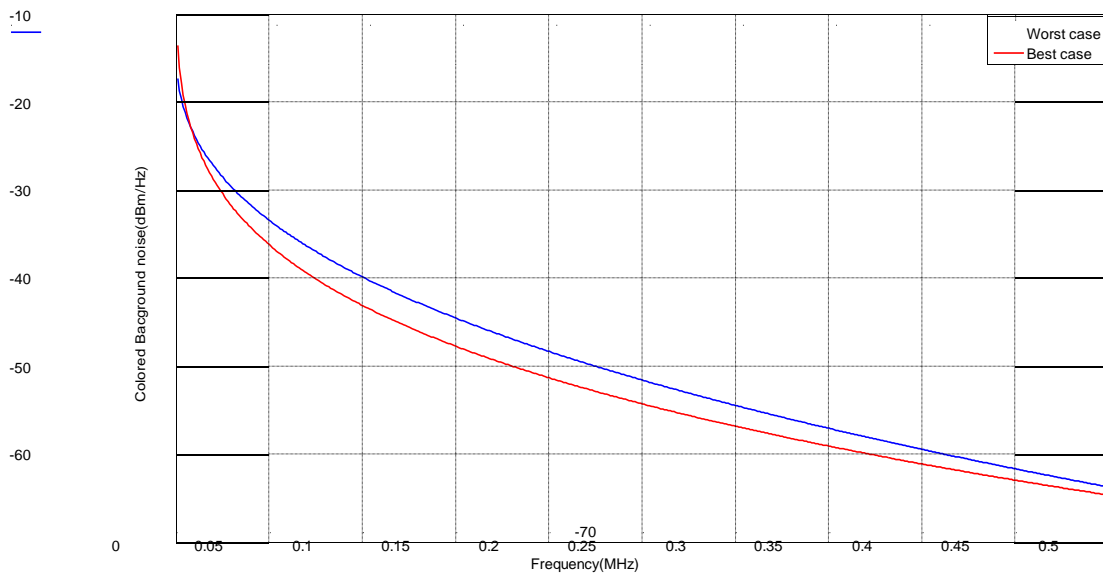


Figure 6. Colored background noise model.

4.2. Narrowband Noise

The narrowband noise is the sum of different narrowband noises. Therefore the proposed model is a summation of a parametric Gaussian function expressed by:

$$N_{NI} = \sum_{i=1}^N A_i \exp\left(-\frac{(f - f_i)^2}{2\sigma_i^2}\right)$$

Where N_{NI} is the number of narrowband interferences, A_i is the amplitude, f_i for the center frequency of each interference and σ_i represents the Gaussian function standard deviation which controls the bandwidth of the narrowband interference. Figure 7 shows the resulting narrowband noise.

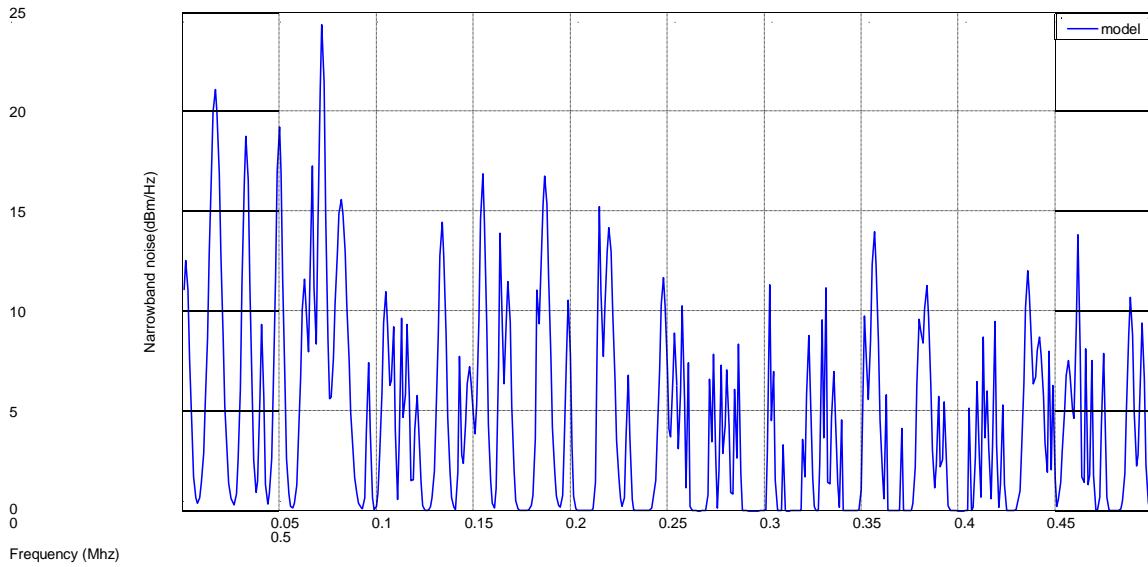


Figure 7. Simulated narrowband noise model.

The parameters of the Gaussian function are derived from measurements by doing the difference between the measured stationary noise PSD and the model of the colored background noise PSD expressed in (1).

When designing a noise model, it is crucial to acquaint the statistical behaviour of the parameters A_i , f_i and σ_i .

Figure 8 illustrates the Cumulative distribution function (CDF) of the narrowband interferences amplitudes A_i ; it is observed that this parameter follows Gamma distribution.

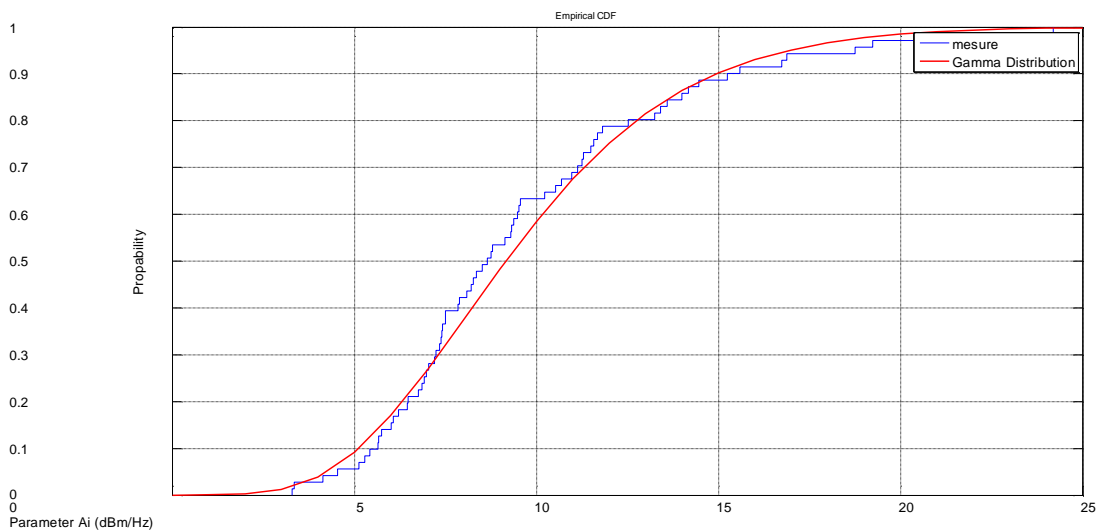


Figure 8. CDF of the parameter A_i

Figure 9, shows the CDF of the narrowband interferences center frequency f_i which is normally distributed.

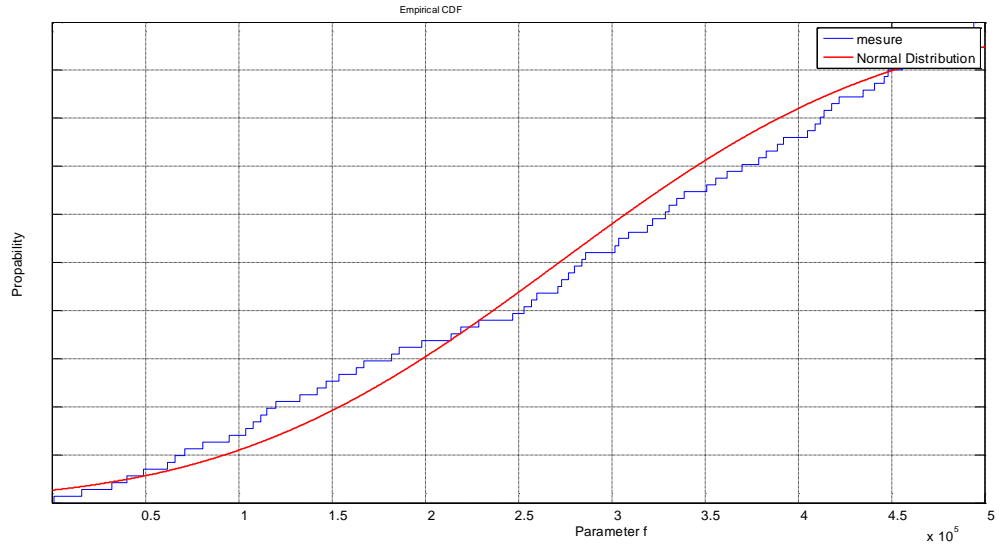


Figure 9. CDF of the parameter f_i

Figure 10 represents the CDF of the parameter σ_i which control the narrowband interferences Bandwidth. It is noted that σ_i follows a Gamma distribution.

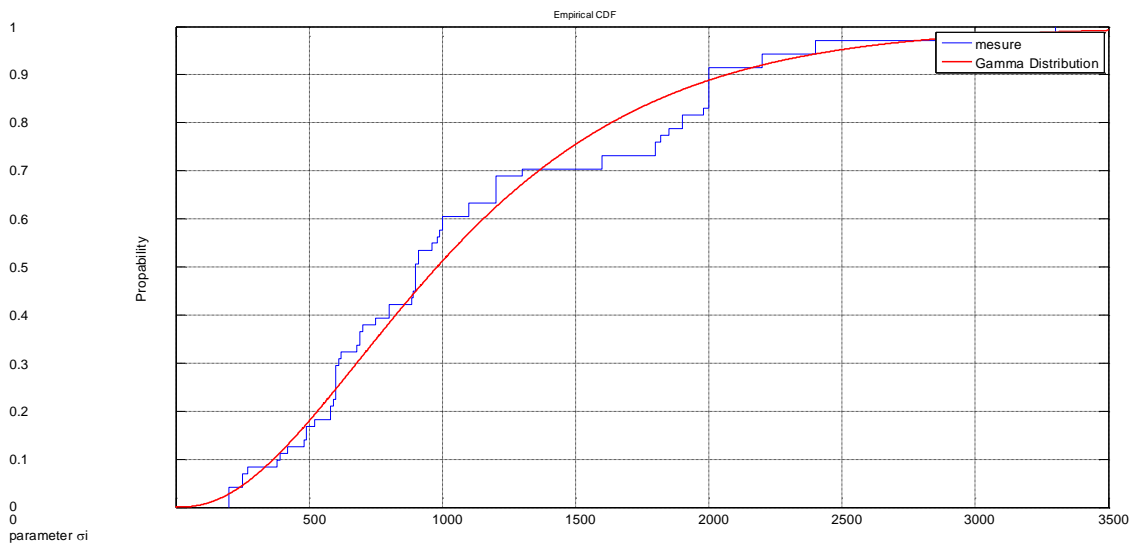


Figure 10. CDF of the parameter

From the statistical analysis of the parameters A_i , f_i and σ_i , we can notice that the narrowband noise model parameters are not unambiguous defined and they have statistical properties. The CDF of the narrowband interferences amplitude A_i follows a Gamma distribution with shape parameter equal to 6.09 and scale parameter equal to 1.58. The parameter f_i is well fitted by a normal distribution with a mean value equal to 2.72 and a standard deviation equal 1.39. The parameter σ_i is also approximated by Gamma distribution with shape parameter equal to 2.54 and scale parameter equal to 440.99.

4.3. Stationary Noise

As it is depicted above, the stationary noise is considered as the superposition of the colored background noise and the narrowband noise because they remain stationary over time. The PSD of the stationary noise is expressed by:

$$N_S = N_{CB} + N_{NI} \quad (3)$$

Where N_S is the stationary noise, N_{CB} is the colored background noise and N_{NI} is the narrowband noise.

Figure 11 illustrates the PSDs of an example of the measured noise and the proposed model.

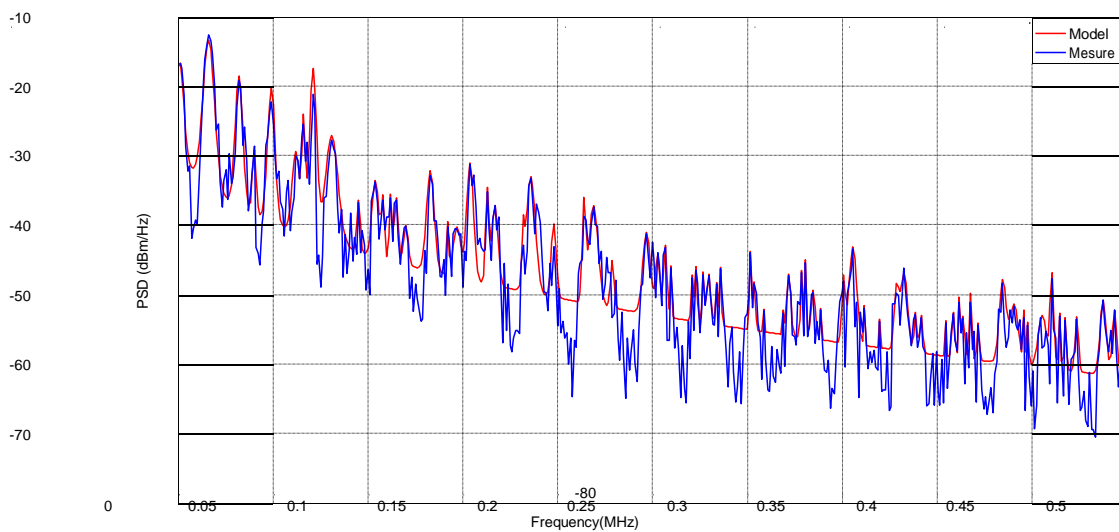


Figure 11. PSD of the Stationary modelled and measured noise.

It is shown that the proposed model is in good agreement with the measured PSD. The red curve shows the stationary noise model following expression (3) and corresponding to the measurement example. We can see the decreasing trend of the background noise with increasing frequencies and the presence of narrowband interferences with higher levels in the very low frequency range.

In order to validate the proposed model we statistically process the simulated noise and compare results with measured noise.

Figure 12 and Figure 13 depict the PDFs and CDFs of the modelled and measured noise amplitude.

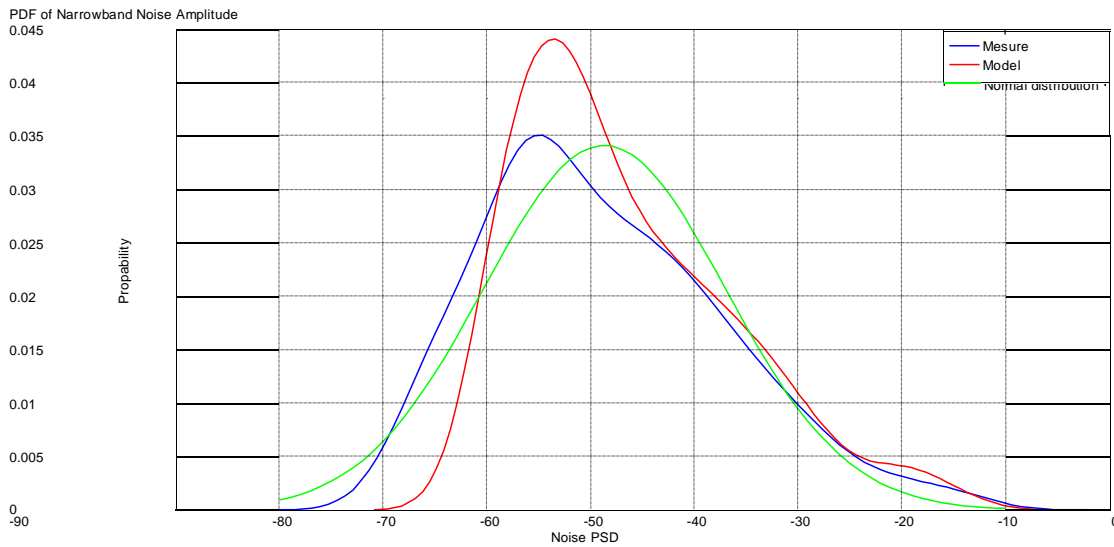


Figure 12. PDF of the modelled and measured noise amplitude.

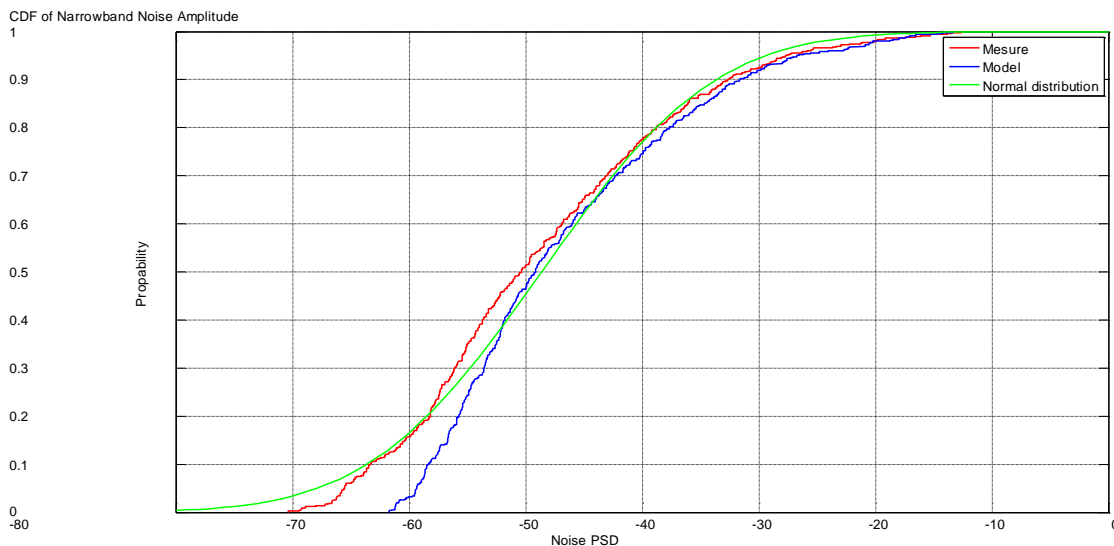


Figure 13. CDF of the modelled and measured noise amplitude.

The comparisons show that the modelled and the measured PDFs are well matched. A similar close match is also achieved for the modelled and measured noise CDF. It is also, observed that simulated noise model has same statistical properties of distribution as the measured noise: In fact, the CDFs of the measured and modelled follow a Gamma distribution with respectively mean value equal to and the Normal distribution fit the PDFs. This proves the relevance of the proposed model.

5. CONCLUSION

In this paper narrowband frequency domain PLC noise is measured and analyzed, and then a statistical characterization and modelling study is presented in detail. Experimental results show that the stationary noise is a decreasing function of frequency with the presence of narrowband interferences. Therefore it is considered as a superposition of the colored background noise and narrowband noise. The colored background noise PSD is modelled by a power function,

however the narrowband noise PSD is considered as a summation of Gaussian functions. The model of the stationary noise is the sum of these two parametric models. In order to validate the proposed model of the stationary noise a basic statistical analysis is given showing the goodness of the proposed model. In future works, the author will be interested to study the narrowband impulsive noise in low voltage outdoor electrical networks. Then, it become easy to simulate the real PLC system and create a complete OFDM PLC communication system and search for the most appropriate coding and modulation techniques.

REFERENCES

- [1] A. Haidine, A. Tabone, and J. Muller, "Deployment of power line communication by European utilities in advanced metering infrastructure," *The 17th International Symposium on Power Line Communications and Its Applications (ISPLC)*, Johannesburg, South Africa, 24-27 March, 2013.
- [2] S. Galli, A. Scaglione, and Z. Wang, "Power line communications and the smart grid", *The first IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, USA, 4-6 October, 2010.
- [3] Dubey, A., et al., "Modeling and Performance Analysis of a PLC System in Presence of Impulsive Noise," *IEEE Power & Energy Society General Meeting (PESGM)*, Denver, CO, USA, 26-30 July, 2015.
- [4] G. Ndo, F. Labeau and M. Kassouf, "A markov-middleton model for bursty impulsive noise: modeling and receiver design," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, October 2013.
- [5] J.A. Cortes, L. Diez, F.J. Canete and J.J.Sanchez, "Analysis of the indoor broadband power line noise scenario", *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 4, November 2010.
- [6] I.Elfeiki, T.Doligez, I. Aouichak, J. Le Bunetel, Y. Raingeaud, "Estimation of PLC transmission line and crosstalk for LV outdoor electrical cables," in *Proceedings of the International Symposium on Electromagnetic Compatibility*, Angers, France, September 2017.
- [7] C. Kaiser, N. Otterbach, K. Dostert, "Spectral correlation analysis of narrowband power line noise," In *Proceedings of the 2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC)*, Madrid, Spain, April 2017
- [8] I. Elfeiki, S. Jacques, I. Aouichak, T. Doligez, Y. Raingeaud and J. Le Bunetel "Characterization of Narrowband Noise and Channel Capacity for Powerline Communication in France," *Energies* 2018.
- [9] Mlynek, P., J. Misurec and M. Koutny, "Noise Modeling for Power Line Communication Model, The 35th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 3-4 July 2012.
- [10] D'Alessandro, S., M. De Pianta and A.M. Tonello, "On Modeling the Sporadic Impulsive Noise Rate within In-Home Power Line Networks," *ISPLC*, Austin, TX, USA, 29 March-1 April, 2015.
- [11] Hirayama, Y., et al., "Noise Analysis on Wide-band PLC with High Sampling Rate and Long Observation Time," *ISPLC*. Tokyo, Japan, 2003.
- [12] Nyete, A.M., T. Afullo and I.E. Davidson, "Statistical Analysis and Characterization of Low Voltage Power Line Noise for Telecommunication Applications," *The 12th IEEE AFRICON*, Addis Ababa, Ethiopia, 14-17 September, 2015.
- [13] Tiru, B., "Analysis and Modeling of Noise in an Indoor Power Line for Data Communication Purpose," *Global Research Publications*, January 2012.

- [14] Esmailian T. "Multi mega-bit per second data transmission over in-building power lines. Doctoral thesis at University of Toronto. 2003.
- [15] OMEGA Deliverable D3.2. PLC Channel Characterization and Modelling. 2008.
- [16] H. Philipps, "Performance measurements of powerline channels at high frequencies, " *The International Symposium on Power Line Communications and Its Applications*, Soka University, Japan, March 1998.
- [17] M. Babic, M. Hagenau, K. Dostert, J. Bausch, "Theoretical postulation of PLC channel model," Open PLC European Research Alliance (OPERA), 2005.
- [18] D. Benyoucef, "A new statistical model of the noise power density spectrum for powerline communication," *International Symposium on Power Line Communications and Its Applications*, Mar. 2003.
- [19] R. Alaya and R. Attia, "Coupling Unit for Narrowband Power Line Communications Channel Measurement," *The 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM*

EDUCATIONAL APPROACH TO THE INTERNET OF THINGS (IoT) CONCEPTS AND APPLICATIONS

Rajeev Kanth¹, Tuomas Korpi¹, Arto Toppinen¹, Kimmo Myllymäki¹,
Jatin Chaudhary² and Jukka Heikkonen²

¹Savonia University of Applied Sciences, Opistotie 2, 70150 Kuopio, Finland

²University of Turku, Vesilinnantie 5, 20520 Turku, Finland

ABSTRACT

The term “Internet of Things (IoT)” and its ecosystem is expanding very rapidly, and therefore, it has become complicated to capture the actual definition of ‘IoT.’ This has created numerous challenges and complications for the students looking forward to achieving accurate perception. This paper is presented with an insight to educate the audiences who do not have sufficient knowledge of the Internet of Things (IoT). The scope of this paper expands from the concepts of Information Technology, Digitalization, Wireless Networking, and sensor technologies to the broad arena of IoT. A simple and easily understandable explanation of IoT including applications and impacts to the society has been proposed. This paper exhibits present state-of-the-art experimental research outcomes as extended examples for the reader to develop a comprehensive conceptualization of IoT. Finally, after going through this paper, a layman who does not have any specific knowledge in this field, do build the concepts and understands the advancements behind the buzzword, “IoT.”

KEYWORDS

Internet of Things; Things-to-Things Connectivity; IoT Applications

1. INTRODUCTION

The dynamicity of the world is vividly increasing with the increment in the smart devices (explained later) and connectivity with them and within them. Reliable and strong connectivity has been possible with sophisticated communication techniques and systems. The peculiarity of recently developed devices stands on their capability of communication within two smart devices and further decision making depending upon the conclusion derived from the communication processes. The introduction of devices capable of communicating within themselves has laid the foundation for the gigantic and phenomenal era of IoT.

In recent years, the buzzword “Internet of Things (IoT)” has gained a lot of popularity within the community of science and engineering. To develop a clear understanding of the field, we studied several pieces of research literature from IEEE, Springer Journals, Elsevier archived databases. The articles hence obtained solely gave a complete understanding of “what is Internet of Things?”, “How it works?” [1], “What will be a simple network architecture of IoT?” [2], “How future world is changing with the application of IoT?” [3], “What could be the possible effects or impacts on human beings in future?” and “How entire world is rapidly adopting the development towards things-to-things connectivity?” [4]. However, we could not get a simple answer to those

above questions within an article, and hence, in this paper, we have presented a better insight towards IoT and solutions to the above problems in a systematic way.

We came across many people around the globe, who try to understand the basic concepts behind the 'IoT', which is a trendy term since the year 2015. If taken into account managers in an academic institutions, or the mid-level officers in an industry, or the persons associated with governmental organizations, or the people related to UN missions, or the workers in a factory,

or the undergraduate and graduate students (whose area of study is not IT or the Computer Science), most of them are unaware of IoT concepts, the phenomena of things-to-things connectivity and its future impacts. This article focuses on this group of audience and attempts has been made to build a vision and develop an insight about IoT, its architecture, it's working, and its future impacts. We also hope that this article will go into a big audience, and it is probable to a considerable number of feedbacks from the readers.

IoT has emerged from the trend of connections and communications between human-to-human, human-to-objects, and objects-to-objects. Practically, this can be taken as a real network of people, things, and devices (intelligent entities that have some processing capabilities). Combining the concept of connectivity along with the practical network through which this connectivity is developed, it can conclude that, this network of communication hence developed is commonly called as "Internet of Things" or IoT.

Taking the definition, a step further, IoT is an extensive range network of multiple devices. IoT devices are generally termed as "smart devices", e.g., smart TVs, smartphones, Smart Refrigerators, etc.. "Smart Devices" can be defined as electronic devices with multiple Sensors and Actuators, possessing the quality of communicating over internet and intranet. These devices also possess the decision-making ability, which takes place by the continuous feedback mechanism and administrator connection. This approach of decision making is termed as "Controlled Decision Making". [5] [6]

An IoT device is a combination of multiple segments, where each segment is considered as a node. The foundation of IoT systems lies in the connections and communications between device-to-device as well as within the device, hence, several entities of different natures, working capabilities, functions, etc. are always communicating with each other. This communication generally takes place by high pace data exchange, and some analysis or interpretation of data being done at every node. Further, this communication and processing of data result in the integration of multiple IoT devices and completion of a more significant task efficiently.

Connected devices are designed for the gathering (by Sensors), transmission (by the established communication network), and processing of information over the network of devices connected to each other. The fusion of different smart devices and processing units into one single device is the fundamental idea behind IoT systems. IoT devices have the power to gather real-time data, process it according to the need, and further transmit the output is a first working loop of an IoT device. The completion of these three essential tasks with speed, accuracy, and reliability makes this technology exceptionally revolutionary for the present era.

The underlying architecture of IoT devices is depicted in Figure 1. This architecture is built depending on the composition and function of an IoT. The different layers of architecture are the Perception layer, Data Management Layer, Network Layer, Business Logic Layer and Application Layer [5]. These layers will be defined in detail with the help of examples in further sections.

2. RELATED WORKS

We have explored and reviewed several articles in order to find a proper definition and concept on the IoT technology, but unfortunately, even after studying many of conference proceedings and journal articles we could only see the architecture, sensing technology, communicating systems, processing units, security and handling of data being focused upon mainly.

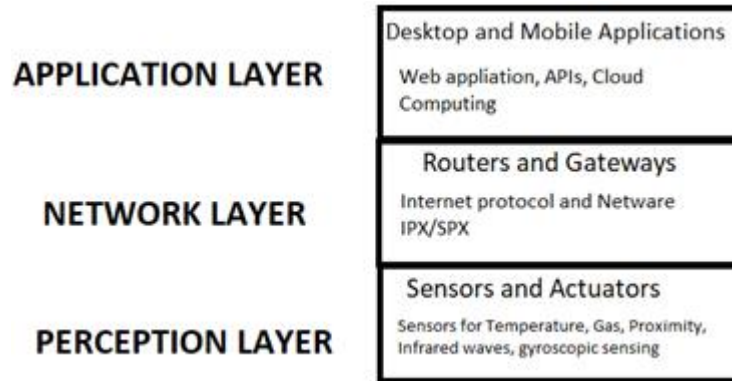


Figure 1. Basic Architecture of IoT devices

The IoT technology is based on the performance of sensors as these equipment are responsible for the human-machine interaction. A complete picture of sensors and their role can be studied in the paper. Though the paper focusses on implementation and importance of sensors in the IoT devices. An application of IoT system is in healthcare for the process of Electrocardiography where sensors play an essential role in collecting the pulses of heart and producing a useful graph. [7]

The data acquisition and transmission after the data collection from the environment becomes the next goal. This is achieved by the combining of the transmission system which is responsible for the transfer of data. Wireless data transmission is the current technology and is widely used which is achieved by various methods of wireless communications [8].

The logic according to which the acquired data will be processed further and an output will be generated is to be tackled. This step is where the integration of computer science with the already existing hardware comes into the picture. The processing of raw data depends on the desired output. The home security system based on the concept of IoT is the state of art technology. The system is programmed such that any unusual behavior sensed by the sensors will be used to trigger the output system where an alert is sent to the owner and an alarm buzzer rings. [9]

The M2M, i.e. machine to machine connectivity, is vital in an IoT system. To achieve reliable connectivity, various radio technologies have emerged, such as Low-Power WiFi, Low-Power Wide Area (LPWA) networks, and various improvements for cellular M2M systems. Intelligent parking systems are an example of an IoT system where a high level of M2M connectivity is implied. In such a system, an IoT device is installed at all the parking locations, and the processing of the image data is done, and an output is generated related to the availability of the parking space in an area. The data needs to be immediately transferred on the cloud so that people looking for parking areas get aware of the availability of parking space. Here, the internet connection over devices was built by an Ethernet connection or a cellular 3G modem. [10]

A large amount of data is gathered by the sensors on a day to day basis. The collection, processing, storing and integration of data from different sensors is required. The data can be Multisource High Heterogeneity Data, Huge Scale Dynamic Data, Low-Level with Weak Semantics Data, and Inaccuracy Data. For the handling of varied types of data, modules such as Data Acquisition and Integration Module, Data Management Module, Data Processing Module,

Data Mining Module, and Application Optimization Module is developed. The management of this massive amount of data from the sensors is done by Data Management Based on Metadata, Semantic Annotations, and Data Indexing Strategy [11].

The complete architecture of the IoT systems for ambient assisted living has been implemented for the study of indoor Air Quality has been conducted. The paper can be concluded with a piece of detailed information about the architecture of an IoT system in general. Here, an IoT system can be seen as a combination of a sensor network, a coordination network, connections with internet, and the output receivers. [12]

We came across scientific pieces of literature that define various parts of IoT with suitable examples, but the deficiency for a paper which explains the concept of IoT, its fundamental working and the flow with which an IoT device is designed in a language that even layman can understand is felt.

3. REFRIGERATOR AND CAR AS IOT DEVICES

After the basic introduction to the IoT and looking at the scope to which IoT can be extended, let us now understand the integrity of IoT in detail and develop an insight about the architectural layers provided in figure 1, in more detail. Let us consider a simple Refrigerator [13], and understand how a refrigerator, which was supposed to be a device for cooling things, can be integrated with the IoT systems, and a “Smart Refrigerator” is developed.

What a refrigerator does, what is its primary function? A traditional refrigerator keeps things (grocery, medicines, water, milk, etc.) cold. A traditional refrigerator is designed just to keep things cold if the door is properly closed. For regular refrigerators, even if the door is opened, it does not give you any signal or beep. This is because the conventional refrigerators do not have the capability of processing the faulty input of the door being opened and the cooling gases escaping out of the door. This can be identified as the very beginning of limitations of conventional devices.

These limitations of conventional devices obstruct it to perform its basic function properly. The introduction to the connectivity of different segments of a device and making the data processing possible at each node has made the functioning very efficient and error-free. In the process of conversion of a conventional Refrigerator to Smart refrigerator, computational intelligence is added to it, so that it can send information when the door is ajar. The intelligence added can also process data regarding, the butter tray being at a low level. By the local resources (not using networked resources, just use local processor capability and database), the refrigerator can send information about whether the food stored in the refrigerator has high-fat content. Moving a step further, by locally programming the electronic integrations, fridge can send information regarding the presence of milk and butter to make a pancake. An analysis of the grocery (by reading the bar code) present inside the fridge can be done and results regarding the possible dish to be prepared can be suggested. So far, the refrigerator is using its own resources and is still not networked with other internet or intranet environments. [14]

Taking the study, a step further, if the refrigerator with computational intelligence is integrated with a WLAN chip and internet environment is created, then the conventional refrigerator can be considered as an IoT device. This IoT device is now capable of conveying information regarding the possible recipe of the dish, which can be prepared from the available grocery, search news and analyze global trends[15]. If it is networked with the internet, it automatically recognizes the butter tray is low and orders a new packet of butter, and the next day you can

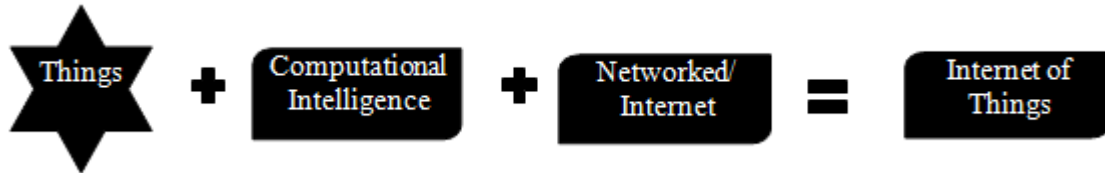


Figure 2. Significant contextual elements of the Internet of Things

collect it to your doorstep. By the above-done integrations, the traditional refrigerator has become smart enough to intelligently read the global trends based on the weather forecast and convey the information regarding the increasing trend in the prices of food items due to approaching winter and further, send a suggestion of buying the grocery beforehand. If not, the refrigerator has got the capability of ordering grocery by itself after performing proper analysis of prices, quality, and food interests of the owner. All these activities of the refrigerator can be controlled in real-time by the administrator by a simple GUI based mobile application. In the other way around, the fridge is capable of providing information about the user's food interest, which will be suitable for the stores, so that the stock of most demanded and less demanded things can be balanced properly.

Evaluating the previous example, we can say that an IoT device must have some intelligence in it and be networked as well. Let us consider a "car." In twenty-first-century a car has a lot of additions than just a vehicle for movement. The modern automobile is a sophisticated combination of sensors, processors, output devices like screens, audio systems, etc. embedded in it. These integrations in the contemporary car have led to features like anti-lock braking system, efficient fuel injection system, better fuel to energy ratio, much more comforting driving experience, and, most importantly, a reliable system for airbag functioning. Apart from the driving experience, the integration of the car with IoT leads to additional amenities like vehicle location, Wi-Fi on board, accident detection GPS based navigation system, etc.. With the current research in Artificial Intelligence domains, the car may be capable of making a memory of driving behaviors and identifying unusual driving behavior and send an SOS notification to the emergency helpline.

The fundamental function of a car to take people from one place to another remains the same, but with the integration of IoT, the transformation is massive. The connectivity to internet has changed and boosted the utilization perspective of consumers. Due to the automation, networked connections and administrator-controlled decision making, the security reliance on the devices has increased up to a vast extent when compared to conventional methods. The internet environment builds within cars provides an opportunity to contact the emergency helpline numbers (for example, 112 in Finland) before or after the crash.

After the examples above, Figure 2 combines the contextual information to a picture and depicts an overview of IoT.

Figure 1 depicted the different layers in an IoT device. These layers are the architectural framework with which any IoT device is modeled and practically realized. Among these, there are three very major layers, namely, Perception Layer, Network Layer, and Application Layer, and these layers can be explained as follows: [16]

- **Perception Layer:** This layer is also known as the Recognition layer. This layer is the lowest layer and its fundamental function is to collect information from the surrounding environment. This layer consists of sensors for the detection of the stimuli being generated in the external environment. These sensors include heterogeneous devices, humidity sensor, temperature sensor, RFID tags, GPS, etc. [17]
- **Network Layer:** It is the brain of an IoT system. The network layer collects the information from the perception layer and facilitates secured data transmission to the application layer for further processing. The entire data processing is taken place at the Network layer and hence, this is called “Core Layer”. Data transmission by encryption processes with unique addresses ensures uninterrupted integration between object-to-object over a single network which is established with the collaboration of numerous devices and hence maintains the universality of the IoT notion. The network layer is achieved by wired, wireless and satellite technologies which include Bluetooth, WiFi, NFCs, etc.[16] [18]
- **Application Layer:** This is the top layer of IoT systems. This layer provides personalized based services according to the user needs. In this layer, the final depth of integration of IT with industry takes place. This layer fills a major gap between user and application and its main function is to share information and maintain the privacy of the data. [16] [19]

Table 1. The different architectural layers in Smart Refrigerator and Smart Cars.

<i>Architectural Layer</i>	<i>Smart Refrigerator</i>	<i>Smart Car</i>
Perception Layer	Bar code reader, door sensors, moisture, and humidity sensor	IR sensors, Actuators, gyroscopic sensors, GPS sensors.
Network Layer	WLAN chip	WLAN chip, Bluetooth connectivity
Application Layer	Mobile application, LCD screen	Buzzer indication, LCD,

4. UNDERSTANDING IOT USING SOME APPLICATIONS

4.1. Smart Learning Environment

Education is the most essential foundation for the human resource of any country. To flourish the economy of any country, the youth needs to be educated and hence, the education sector still persists to be the most focused for any government. Proper education is one of the major foundations of our society and needs to be accessible to everyone. The conventional education system where the teacher interacts with the student and vice versa has brought up a lot of challenges, among which the prominent ones are classroom environment, lack of visualization, absence of practice tests, etc.. The considerable decrement in the understanding levels of the students has been a significant challenge to the research community.

IoT systems have helped in tackling the challenges being faced by the conventional teaching methodology and has proven to make learning more comfortable, more individual, and more effective. Figure 4, depicts the IoT based learning system. The diagram shows the modern approach to a teaching-learning process where an IoT based environment is created. The central part of this environment is the server upon which the entire concept lies. This server connects to the devices of students and teachers via an interactive computer-based application, and this server is connected to other devices via internet. This set-up can be termed as Smart Learning Environment (SLE). Smart Learning Environments (SLEs) shall help to establish a seamless connection between a virtual and a physical environment. SLEs adjust content and studying techniques according to the need of the student and provide a platform to communicate with others. The interaction with teachers can be done easily over the server and moreover, by the close monitoring of the learning patterns of students, teachers can take care of every student in a unique way. The lack of visualization among students can be improved by the use of 3D- printers, visualization tools, etc.. In this way, the teaching will be depictive and demonstrative rather than the conventional lecture method. Moreover, it is seen that the physical environment of the classroom affects the teaching and learning process. The temperature, ventilation, mic, presentation board, etc. affect the leaning process [20]. These factors can also be controlled by the server by building intranet-based connectivity. The server can be connected with the temperature sensor and air conditioner both, the motors in the windows can be connected with the server, the mic and the amplifiers can be connected together with the server, etc. In a similar way, all the other internal devices (i.e. devices inside the lecture hall) can be connected to the server via intranet connectivity, and an administrator-based control can be gained over the external factors of the hall. With the development of digital classrooms, the standard of education being imparted can be made efficient as the units in a digital classroom works on the coordination of IoT devices which are interconnected and works on the sensors, sensor data and is further, processed by using Artificial Intelligence algorithms. [21][22]

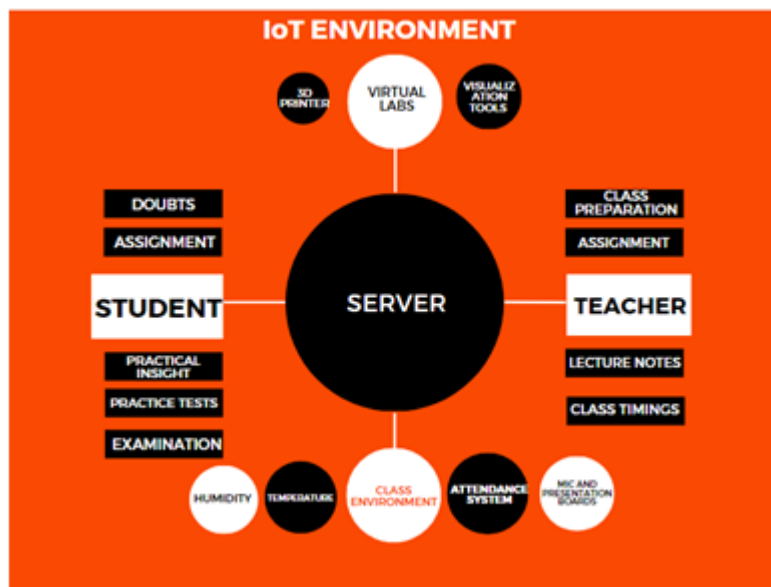


Figure 4. The IoT based modern classrooms.

4.2. Disaster Control

The occurrence of a disaster in a state is an emergency where there is a lot of effort and promptness needed from the rescue team to tackle the challenge posed. The main goal during the time of disaster is to save the lives of citizens by robust planning and execution within least time

and minimal failure risks. According to Quaeantelli (1988), the major problem faced by the rescue team during disaster management is in coordinating among the team in the rescue process [23]. In such critical scenarios where the life of citizens are at stake, the integration of IoT with safety equipment has proved to be reliable. IoT technologies help to recognize life-threatening dangers, warn citizens and evacuate them as soon as possible. They can also support the rescue workers by providing them with information and hence, help to tackle the problem of poor coordination faced at different steps among the rescue workers. A wireless sensor network can be created which transmits the information regarding temperature, humidity, etc. along with the development of a communication path over radio waves [24]. Figure 5 shows the block diagram of the wireless sensor network which is a novel method of talking about the problem of poor communication and coordination.

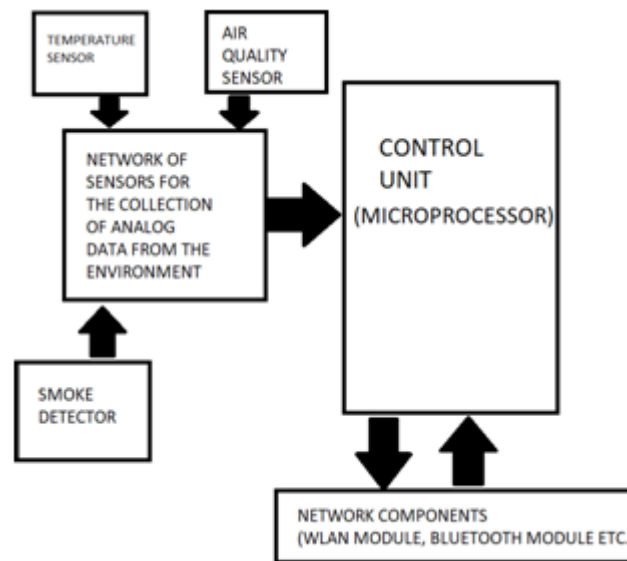


Figure 5. Block diagram showing the connections between microcontroller and other sensors as a part of the architecture of the device.

In the case of jungle fire, prior installment of sensors on trees, monitor different parameters to warn as fast as possible of wildfire and alarm the local fire department and the team for fire extinguishing can act robustly. Moreover, by a simple mobile application, the citizens of the place can be warned of the fire and further, safety measures can be taken. IoT can be a life savior in the case of building and factory fires as well. In cases of a building fire, the wireless sensor network is being utilized to automatically stop lifts, open emergency exit gates, glow all the emergency signals and fire alarms to warn all the people stuck inside the building, whereas robots and machines in factories are stopped immediately and fire extinguishers are activated. The promising accuracy and precision which IoT devices provide are being utilized for the management of disaster widely and get control over it. These tactics are also used to save the citizens from terror attacks, earthquakes, electric breakdowns, etc.. IoT innovation is still developing in this arena with utmost pace to take make the system fully automatic and reliable [25].

Floods have been known for creating a huge number of casualties and infections due to the waterlogging and intake of impure water [26][27]. The maintenance of fresh drinking water and disintegrating the water logging during the flood is the biggest challenge for the government and as a failure to achieve it, diseases like diarrheal infections, acute respiratory infections, malaria, leptospirosis, measles, dengue fever, viral hepatitis, typhoid fever, meningitis, as well as tetanus and cutaneous mucormycosis has been known to spread during flood. [28] These diseases occur

because of the increase in the disease vector which takes place due to complex sources at the sight of flood.

Prevention over the water prone diseases spreading in the areas of waterlogging, eventually leading to increment in the disease vector during a flood can be easily done by IoT technology.

In such scenarios, a system of sensors detecting the type of microorganisms growing in the water is installed and the data received from these sensors is transmitted to government bodies so that proper treatment and precautions can be taken at the place [29]. Moreover, the citizens are informed regarding the current situation and the water quality and water levels can be monitored by the government agencies. The healthcare facilities and supply of antidote of the pathogens attacking the population at the site of calamity is done, leading to a reduced number of casualties. Similarly, in the tropical regions of the world, heavy snowfall can lead to huge casualties and is considered to be a great disaster. To tackle this, an IoT based drone system has been constructed, which measures the depth of the snow coagulated. The information gathered by the RADAR of the drone is processed and published on the cloud platform so that residents of such places can be protected [30].

4.3. Smart Homes

The advancement in IoT technology has led to the ease in the maintenance of smart gadgets inside the house. Home automation has become very popular as it becomes difficult to manage household gadgets in busy routines. The quality of life has been increased up to a huge extent due to the automation provided by IoT devices. The connectivity of devices-to-devices has led to the ease in the operation of gadgets. E.g., the lighting systems of the different rooms can be controlled by the mobile application [31]. This kind of system is considered to be partial automation as the intervention human being is required whereas cases where there is no human intervention required leads to a fully automated system. E.g., by advanced algorithm, the air-conditioner calculates the time taken to drive the room's temperature to the user's comfortable temperature and gets started automatically before the user arrives home from the office with the perfect setting of temperature. It should be noted that in both the conditions, the control exists with the user and the system can be stopped/modified according to the desires of the user.

4.4. Smart Parking Management System

Intelligent parking management was a huge challenge for the road authorities, and it was posing great difficulties to the drivers. In big cities, finding an empty parking spot was a massive task as the management was hazy and unclear. IoT technologies gave a solution to this problem with the integration of internet and RFID tags and readers. In this technological advancement, all the parking spots are connected to the internet via RFID tags and the information of the place being filled or empty is transmitted over the internet based on the output of these RFID tags. Moreover, when a car arrives at the parking spot, the car's RFID tag is read by sensors, and the time for which the car was parked is billed and the bill is sent to the owner [32].

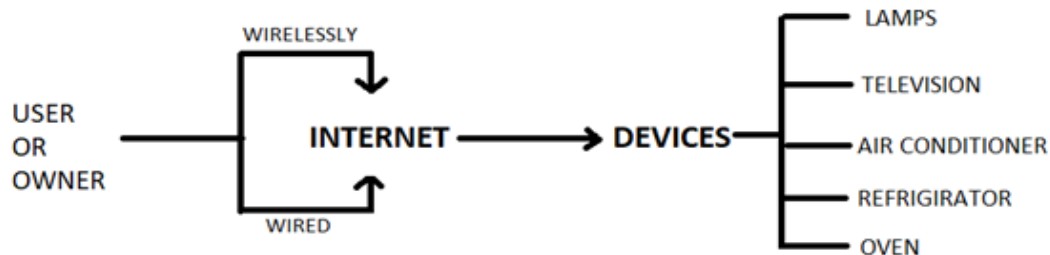


Figure 6. The simple flow chart for the working of the home automation devices.

4.5. Titanic vs Amorella

The advancement in IoT can be felt by comparing two cases from the past. The first case is of the Titanic ship, which sank in the water after hitting the iceberg in 1912. After thorough studies

by different scientists, it was found that one reason behind a considerable number of casualties in the accident was the lack of connectivity and communication among ships in the same area. It is also believed that a lot of people could have been saved if the information about the accident of Titanic was shared among the other ships and the control unit instantly [33]. A similar incident took place with a passenger ship, Amorella, which ran aground while it was on a voyage from Finland to Sweden in the Baltic Sea in December 2013. Since Amorella ship was equipped with the latest emerging technologies of connectivity and a well-established GPS network, lives of all the passengers, including crew members were rescued by sending a rescue team on time.

5. A DETAILED EXAMPLE: IOT SMART BIN

Technology is evolving rapidly, which makes the price and size of batteries and microchips smaller while computing power and battery capacity is increasing. This evolution has made it possible to develop a new kind of information technology network capable of exchanging data and communicating with each other in real-time, using programmable microchips and sensors connected to the internet, radio networks or telecommunications network. These technologies enable people to interact remotely with the real world. Cloud computing makes it possible to process a massive amount of data generated by these networks to analyze it almost instantly.

In this regard, we have designed and developed an IoT based smart bin that is capable of transmitting data to the cloud in real-time based on load sensor readings. This experiment was carried out in the Electronics Laboratory of Savonia Institute of Applied Sciences, Finland as a part of orientation project. A prototype for the IoT bin was built that measures the weight of the container and the surrounding temperature and sends the data to the required authorities. Four load cell sensors were used that handle 50 kg each, so maximum weight allowed is 200 kgs. Load cells were connected, using Wheatstone bridge, to an analog to digital converter (HX711) that converts pressure on the sensors to voltage calculated in the microchip. The microchip used was ESP8266 WLAN module-based chip (ESP8266 NodeMCU). It analyses the information coming from the converter and also sends it to a website by connecting to the internet and cloud services. All these tasks are programmed into a microchip using Arduino IDE software. As a

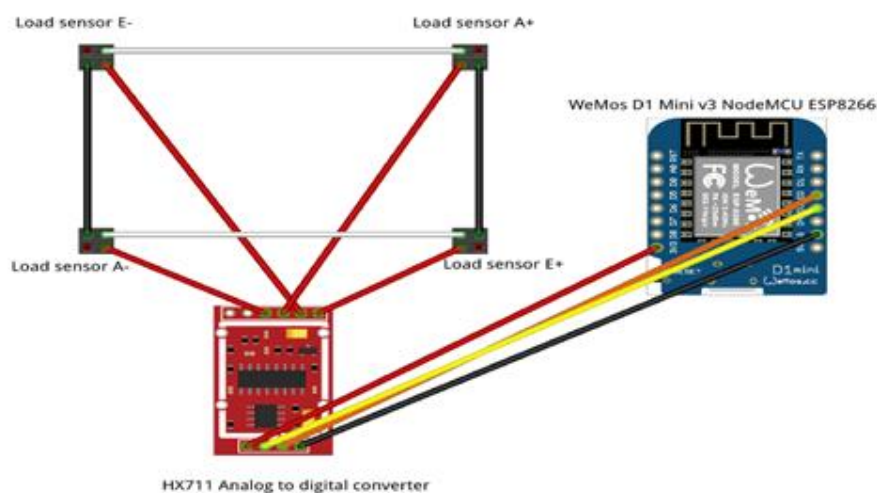


Figure 7. Wiring and Setup of the Components

cloud platform, Google Firebase is used. Firebase provides NoSQL real-time database with website hosting, so data is updated almost instantly to the webpage where it is visualized and plotted into a graph. Data can be accessed with a computer or a mobile phone using an internet

connection. ESP8266 board is powered with the rechargeable battery through a micro USB port. It is also possible to power them with necessary mobile phone charger using AC, as some of the waste bin shelters have electricity for lights and hence, these AC ports can be used for power the ESP8266 board.

Firebase cloud services provide hosting only, so the website still has to be built from scratch. A functional website for the prototype, as shown in Figure 8, was created using basic HTML, CSS, and JavaScript. Firebase is free to apply for a smaller volume of data to be displayed. If it is necessary to store and view more than 1 GB of data per month, it should be updated using a paid version.

The prototype as shown in Figure 8. acted as a proof of concept that it would be possible to monitor waste weight remotely and possibly send a bill based on the load of the bin to the client. However, there are still some revisions to be made before it would be possible to implement into a waste management process. For proper functioning, wireless internet is required by the developed system. It would be cost-effective to provide region-wise wireless internet for the developed system, or it could be used only in the areas where there is a condominium owned internet service. During the wintertime, battery-powered scales would not be an option if bins were kept below negative temperatures. The implementation of this technology can help the person who is responsible for the monitoring of the garbage levels and clean it regularly, especially to the driver of garbage container vehicles. As a future prospect, the data obtained from the sensors can be further analyzed to achieve the real-time graphs related to the filling rate, micro-organisms being developed in the bins. A ratio of biodegradable to non- biodegradable wastes can also be calculated. These data will be helpful in further studies on waste disposal by the government authorities.

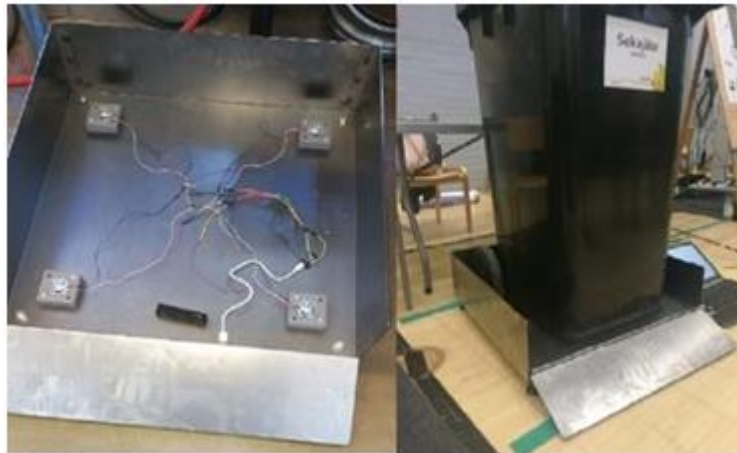


Figure 8. Final Working Prototype

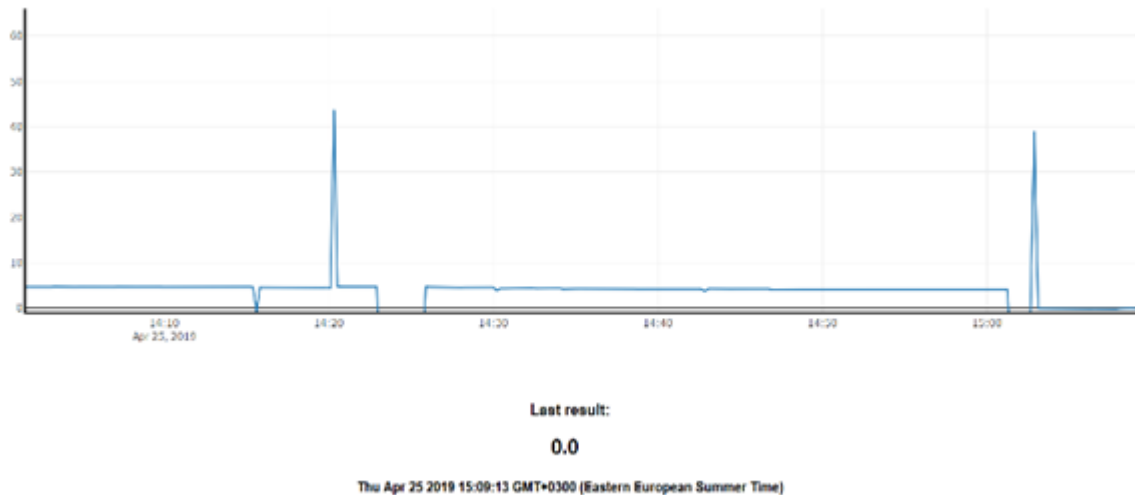


Figure 9. Screen capture of data visualized and updating instantly to the webpage

6. HOW THE WORLD IS CHANNING WITH THINGS CONNECTIVITY AND FUTURE IMPACTS

IoT is the technology of the future. Scientists across the globe are working for the integration of different devices. The GSMA intelligence [34] report reveals that mobile connections, including licensed cellular IoT, has exceeded beyond 9.32 billion subscriptions. The earlier days' station to station phone connectivity has been changed to the station to people and then people to people and now an era of a device to device connectivity is at our doorsteps. This significant change in the connectivity entirely shifts the principles, technologies, systems, and applications of our daily used gadgets like mobile phones, television, refrigerator, etc. and hence, the most popular term "SMART DEVICES" has evolved. The acceleration in the integration of IoT technology with different areas of Science has been taking place at a very high rate.

There exists a vast range of scientific research scopes possible in this field. Proximately, IoT is expected to lead to the development of an earthquake detecting device with the integration of principles of IoT and earth sciences. This device will be capable of sensing the activities of the inner earth before the earthquake and give signals to the control unit outside. Further, the replacement of medical practitioners with highly efficient robots is expected to be seen very soon, and the security system can be completely automatized by the integration of multiple sensors, memory units, and advanced processing power, etc..

7. CONCLUSION

This paper was targeted to the people who aren't associated with the field of computer science and Electronics and has no knowledge of IoT. The objective of the paper was to present the idea of the topic 'Internet of Things' in a straightforward way so that even a layman can understand it after reading this article. The topic was explained by proper examples and real-life implementation of IoT. To give the reader a picture of the state of art development in the field, the experimental work on the monitoring system of garbage bins was presented in the later stage of the paper in greater detail and proper discussion. This experiment shows how a simple sensing technology can be combined with cloud computing and innovative technology can be developed.

REFERENCES

- [1] P. J. Rani, J. Bakthakumar, B. P. Kumaar, U. P. Kumaar and S. Kumar, "Voice-controlled home automation system using Natural Language Processing (NLP) and Internet of Things (IoT)," 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai, 2017, pp. 368-373. doi: 10.1109/ICONSTEM.2017.8261311.
- [2] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun and Hui-Ying Du, "Research on the architecture of Internet of Things," 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, 2010, pp. V5-484-V5-487. doi: 10.1109/ICACTE.2010.5579493.
- [3] Coetzee, Louis, and Johan Eksteen. "Internet of things—promise for the future? An Introduction." (2011).
- [4] S. Agrawal and M. L. Das, "Internet of Things — A paradigm shift of future Internet applications," 2011 Nirma University International Conference on Engineering, Ahmedabad, Gujarat, 2011, pp. 1-7. doi: 10.1109/NUiConE.2011.6153246.
- [5] T. Fan and Y. Chen, "A scheme of data management in the Internet of Things," 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content, Beijing, 2010, pp. 110- 114. Doi: 10.1109/ICNIDC.2010.5657908
- [6] R. Khan, S. U. Khan, R. Zaheer and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," 2012 10th International Conference on Frontiers of Information Technology, Islamabad, 2012, pp. 257-260. Doi: 10.1109/FIT.2012.53
- [7] S. K. Dhar, S. S. Bhunia and N. Mukherjee, "Interference Aware Scheduling of Sensors in IoT Enabled Health-Care Monitoring System," 2014 Fourth International Conference of Emerging Applications of Information Technology, Kolkata, 2014, pp. 152-157. DOI: 10.1109/EAIT.2014.50
- [8] Godavarthi, Bhavana, Paparao Nalajala, and L. R. Teja. "Wireless sensors based data acquisition system using a smart mobile application." *Internet of things, "International Journal of Advanced Trends in Computer Science and Engineering* 5, no. 1 (2016): 25-29.
- [9] A Anitha, IOP Conf. Series: Materials Science and Engineering 263 (2017) 042026, DOI:10.1088/1757-899X/263/4/042026.
- [10] S. Andreev et al., "Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap," in *IEEE Communications Magazine*, vol. 53, no. 9, pp. 32-40, September 2015. DOI: 10.1109/MCOM.2015.7263370
- [11] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges," in *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75-87, Feb. 2017. DOI: 10.1109/JIOT.2016.2619369
- [12] Marques G, Pitarma R. An Indoor Monitoring System for Ambient Assisted Living Based on Internet of Things Architecture. *International Journal of Environmental Research and Public Health*. 2016; 13(11):1152. <https://doi.org/10.3390/ijerph13111152>
- [13] Online Lecture Series on the Internet of Things by Prof. Ian G. Harris at UCI University of California, Irvine, Coursera online learning platform.
- [14] A. Floarea and V. Sgârciu, "Smart refrigerator: A next generation refrigerator connected to the IoT," 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, 2016, pp. 1-6. doi: 10.1109/ECAI.2016.7861170.
- [15] S. Luo, H. Xia, Y. Gao, J. S. Jin and R. Athauda, "Smart Fridges with Multimedia Capability for Better Nutrition and Health," 2008 International Symposium on Ubiquitous Multimedia Computing, Hobart, ACT, 2008, pp. 39-44. doi: 10.1109/UMC.2008.17
- [16] Bilal, M. (2017). A review of internet of things architecture, technologies and analysis smartphone-based attacks against 3D printers. arXiv preprint arXiv:1708.04560.
- [17] H. Suo, J. Wan, C. Zou and J. Liu, "Security in the Internet of Things: A Review," 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, 2012, pp.648-651. Doi: 10.1109/ICCSEE.2012.373.
- [18] Sagar Shriram Salwe, Karamtot Krishna Naik. (2019) Heterogeneous Wireless Network for IoT Applications. *IETE Technical Review* 36:1, pages 61-68.
- [19] Yun, M., & Yuxin, B. (2010, June). Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. In 2010 International Conference on Advances in Energy Engineering (pp. 69-72). IEEE.

- [20] Suleman, Q., Aslam, H. D., & Hussain, D. I. (2014). Effects of Classroom Physical Environment on the Academic Achievement Scores of Secondary School Students in Kohat Division, Pakistan. *International Journal of Learning and Development*, 4(1), 71. <https://doi.org/10.5296/ijld.v4i1.5174>
- [21] Hanan Aldowah, Shafiq UI Rehman, Samar Ghazal, Irfan Naufal Umar, "Internet of Things in Higher Education: A Study on Future Learning", Published in *Journal of Physics, Conference Series*, Vol: 892, DOI: 10.1088/1742-6596/892/1/012017
- [22] Rajeev Kanth, Mikko Jussi Laakso, Paavo Nevalainen, Jukka Heikkonen, "Future Educational Technology with Big Data and Learning Analytics", Published in 2018 IEEE 27th International Symposium on Industrial Electronics (ISIE), Cairns Australia, PP: 906-910, DOI: 10.1109/ISIE.2018.8433753
- [23] Quarantelli, E. L. (1988). Disaster Crisis Management: A Summary Of Research Findings. *Journal of Management Studies*, 25(4), 373–385. doi: 10.1111/j.1467- 6486.1988.tb00043.x
- [24] Ismail, M. N., Shukran, M. A., Isa, M. R. M., Maskat, K., & Adib, M. Establishing a IoT Wireless Sensor Network (WSN) Communication for Peacekeeping Operation.
- [25] Vijayalakshmi, S. R., & Muruganand, S. (2017). Internet of Things technology for fire monitoring system. *Int. Res. J. Eng. Technol*, 4(6), 2140-2147.
- [26] Dewan, T. H. (2015). Societal impacts and vulnerability to floods in Bangladesh and Nepal. *Weather and Climate Extremes*, 7, 36-42.
- [27] Hakim, S. T., Afaque, F., Javed, S., Kazmi, S. U., & Nadeem, S. G. (2014). Microbial agents responsible for diarrheal infections in flood victims: a study from Karachi, Pakistan. *Open Journal of Medical Microbiology*, 4(2), 106.
- [28] Kouadio, I. K., Aljunid, S., Kamigaki, T., Hammad, K., & Oshitani, H. (2012). Infectious diseases following natural disasters: prevention and control measures. *Expert review of anti- infective therapy*, 10(1), 95-104.
- [29] Sinha, A., Kumar, P., Rana, N.P., Islam R, Dwivedi Y. K., "Impact of the internet of things (IoT) in disaster management: a task-technology fit perspective", Published in *Annals of Operations Research* (2017). <https://doi.org/10.1007/s10479-017-2658-1>
- [30] Henry Tarvainen, Eemeli Tolppanen, Petri Selkivaara, Rajeev Kanth, Arto Toppinen, Jukka Heikkonen, " Measurement of Snow-depth using Frequency Modulated Continuous Wave Radar Sensors", Published in 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA). In press, RG DOI: 10.13140/RG2.2.28342.96327
- [31] R. Piyare and M. Tazil, "Bluetooth based home automation system using cell phone," 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore, 2011, pp. 192-195. doi: 10.1109/ISCE.2011.5973811
- [32] L. Chou, C. Sheu and H. Chen, "Design and Prototype Implementation of A Novel Automatic Vehicle Parking System," 2006 International Conference on Hybrid Information Technology, Cheju Island, 2006, pp. 292-297, doi:10.1109/ICHIT.2006.253626
- [33] Kozak-Holland, Mark. *Titanic lessons for IT projects*. Multi-Media Publications Inc., 2005.
- [34] Definitive data and analysis for the mobile industry, <https://www.gsmaintelligence.com/>, last accessed on 1st November 2019

AUTHORS

Dr. Rajeev Kanth was born in Rajbiraj, Nepal, on July 29, 1971. He received a Doctor of Science (D.Sc.) in Information and Communication Technology from University of Turku, Finland, in 2013 and a docent title in 2019. He is currently working as a Senior Lecturer at the Savonia University of Applied Sciences, Finland where he is focusing on teaching and research on Industrial Internet of Things (IIoT). Previously, he has worked at the Indian Space Research Organization (ISRO), Ahmedabad India, Royal Institute of the Technology (KTH), Stockholm, Sweden and the University of Turku (UTU), Finland, where he has been a Researcher, Post-doctoral Researcher, and the Senior Researcher respectively. His current research interests include image and video analysis, Internet of Things, Big Data Analytics, and Artificial Intelligence. He has published more than 45 scientific articles in peer-reviewed conference proceedings and refereed journals in the field of computer science and communication technology. He has been a member of IEEE communication society, IEEE cloud computing community, IEEE Earth Observation Community, and green ICT community.



Tuomas Korpi was born in Finland and currently, he is pursuing a bachelor's degree in Internet of Things (IoT) from Savonia University of Applied Sciences, Finland. He has a keen interest in developing innovative and novel applications of Internet of things using Arduino and Raspberry pi platforms with the suitable sensors.



Arto Toppinen has worked in industrial and public Projects as Researcher, Teacher and Project Manager for over 32 years. Arto Toppinen has extensive experience in wireless and RF Product Development. He has been in the front Edge in developing Sensors to Pulp and Paper Industry. He has a wide Spectrum of technical Know-How based on his Teacher Career in Savonia University of Applied Science. Arto Toppinen has gained deep Experience in quality Management in R&D of wireless Industry. He has a good international Network of R&D Partners in several Countries and universities. He has established three Spin-Offs of IT-firms, ERP-Competech Oy, APL Systems Oy, and Thinking Lifeline Oy



Kimmo Myllymäki was born in Finland. He has achieved a degree of Master of Engineering and holds a vast working experience at Nokia Networks in both Finland and in the UK for about 25 years. While working in Nokia Networks, he was working in many different roles in base station production and R&D departments. Currently, he is working as a dean and manager for electrical engineering and ICT teams at Savonia University of Applied Sciences, Finland. His interest includes Radio Frequency applications, mobile networks, antenna design and Internet of Things applications.



Jatin Kumar Chaudhary was born in Nepal in 1998 and is currently a 3rd-year Electronics Engineering student studying at Sardar Vallabhbhai Patel National Institute of Technology, Surat India. He is very active in research, and his research interests include Solar Cell Optimization, Artificial Neural Network, and the Internet of Things. He has published over four peer-reviewed articles in the reputed scientific forums and in the conference proceedings.



Jukka Heikkonen has been a professor of computer science at the University of Turku, Finland, since 2009. His current research as the head of the Algorithms and Computational Intelligent (ACI) research group at the University of Turku is related to intelligent and learning systems, especially including machine learning, probabilistic, and information-theoretical modeling issues applied in wide varying application domains. He has worked at top-level research laboratories and the Center of Excellence in Finland and international organizations (European Commission, Japan) and has led many international and national research projects. He has authored more than 150 scientific articles.



SECURITY FRAMEWORK FOR IOT DEVICES AGAINST CYBER-ATTACKS

Aliya Tabassum and Wadha Lebda

Department of Computer Science and Engineering, Qatar University,
Doha, Qatar

ABSTRACT

Internet of Things (IoT) is the interconnection of heterogeneous smart devices through the Internet with diverse application areas. The huge number of smart devices and the complexity of networks has made it impossible to secure the data and communication between devices. Various conventional security controls are insufficient to prevent numerous attacks against these information-rich devices. Along with enhancing existing approaches, a peripheral defence, Intrusion Detection System (IDS), proved efficient in most scenarios. However, conventional IDS approaches are unsuitable to mitigate continuously emerging zero-day attacks. Intelligent mechanisms that can detect unfamiliar intrusions seems a prospective solution. This article explores popular attacks against IoT architecture and its relevant defence mechanisms to identify an appropriate protective measure for different networking practices and attack categories. Besides, a security framework for IoT architecture is provided with a list of security enhancement techniques.

KEYWORDS

Attacks, Architecture, Internet of Things (IoT), Intrusion Detection System, Security.

1. INTRODUCTION

In recent years, the number of smart IoT devices has increased dramatically. Due to the cheaper costs of hardware and open-source software, various companies are manufacturing IoT devices. A report published by HP, as a part of the Open Web Application Security Project (OWASP), proves that manufacturers ignore security aspects while developing these devices [1]. Hence, IoT devices have become potentially vulnerable targets for cybercriminals. In addition, it has become difficult for security specialists to secure the huge amount of data residing on the devices and the data in transmission in IoT networks. The complexity due to the number of IoT devices and networks provide opportunities to hackers to turn simple devices like TVs, cameras, DVDs and hubs into harmful botnets to launch jeopardizing cyberattacks [2]. To incorporate major security solutions such as cryptography in IoT devices there are two major challenges: (1) disestablished architecture, infrastructure and standards (2) unsupportive and insufficient resources. Applying appropriate defence mechanism (mitigation) is necessary to block the adversaries to reduce impact on the devices and/or end-users. Although the ever-increasing attacks are difficult to be mitigated fully, real-time network monitoring using an Intrusion Detection and/or Prevention system and adoption of strong access control & authentication mechanism can prevent attacks. The goal of our research is to provide detailed analysis of types of existing defence mechanisms for various attacks detection. So, that the most appropriate approach suitable to the current IoT networking is identified. In this paper, we explore persistent attacks against IoT devices and networks. After which, we provide details on current trends of security mechanisms that are being

adopted to secure IoTs against such attacks. Further, we deduce the future deterministic metrics of IDS after a precise study of various IDS developments in literature. Lastly, from the analysis and review, we suggest a robust framework for securing IoT devices. The structure of the paper is as follows: section 2 provides background and overview on IoT devices, followed by the recurrent attacks against IoT architecture and various security mechanisms developed by security experts, in section 3. Section 4 elaborates on the types of significant security mechanisms that are potential in securing heterogeneous IoT networks. Later, section 5 recollects the crucial security mechanisms and a security framework. Finally, section 6 concludes the work.

2. BACKGROUND

IoT is an interconnection of billions of heterogeneous objects through the Internet. The number of connected smart IoT devices have surpassed the human population and in 2018, the number reached 7 billion. Moreover, researchers predict that in 2025 this number may peak to 22 billion with expected economy generated by various application domains is 4 to 11 Trillion Dollars [3] [4]. Figure 1 shows various application areas of IoT devices, which includes Smart Grid, Smart Retail, Smart Supply Chain, Smart Agriculture, Smart Industry, Smart Transportation, Smart Health, Smart Wearables, Smart Housing & Buildings and Smart City. From the mentioned statistics and areas of application, it is clear that IoTs are present in almost every sector and so, it has become essential to know how an IoT device works.

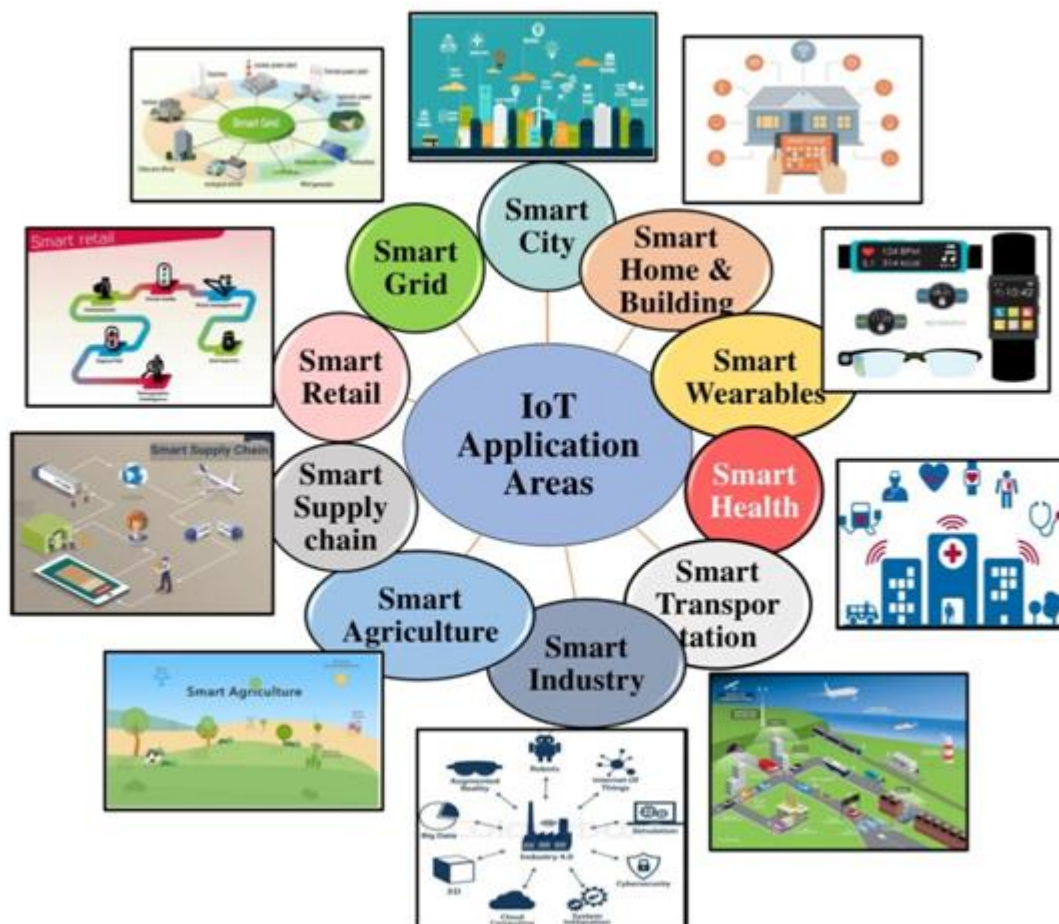


Figure 1. IoT application areas

Any IoT device operates in 3 phases: Collection Phase, Transmission phase and Processing, Management, Utilization phase.

- **Collection Phase:** It is the initial step to collect data from the physical environment using short-range communication sensing devices and technologies [5]. The devices for this phase have less battery power, limited memory and processing power. The design of the communication protocols is in such a way that it consumes less energy, operates on limited data rate, small memory and processing power for short distances. Because of the above reasons, these networks are referred to as Low-power and Lossy networks (LLNs). Consequently, the security mechanisms must be adaptable to the resource constraints of these devices.
- **Transmission phase:** This phase transmits the data collected from the Collection phase to the users and applications using transmission technologies such as Ethernet, Wi-Fi, Bluetooth, Bluetooth Low Energy (BLE), Hybrid Fiber Coaxial (HFC) and Digital Subscriber Line (DSL) [6]. Most of these technologies are vulnerable to attacks. Gateways integrate LLN protocols employed in the collection phase with the Internet protocols of transmission phase.
- **Processing, Management, Utilization phase:** The applications of this phase processes the collected data to get information about the environment. Sometimes, the applications have to make decisions based on the collected information [7]. It also has a middle-ware to integrate the communication with physical objects and multi-operation applications.

The above phases of operation need protection to ensure appropriate delivery of services. In the next section, we explore recurrent security attacks against IoT architecture.

3. ATTACKS AGAINST IOT LAYERS

Although there is no standardized model of IoT architecture, the basic types of architectures that are popularly used are 3 layers, 4 layer and 5 layer architectures and the recent advancements have more abstract layers added to these [8]. In our article, we explore the attacks in three layers, Perception, Transport and Network, shown in Figure 2 as these layers are highly targeted by security attacks.

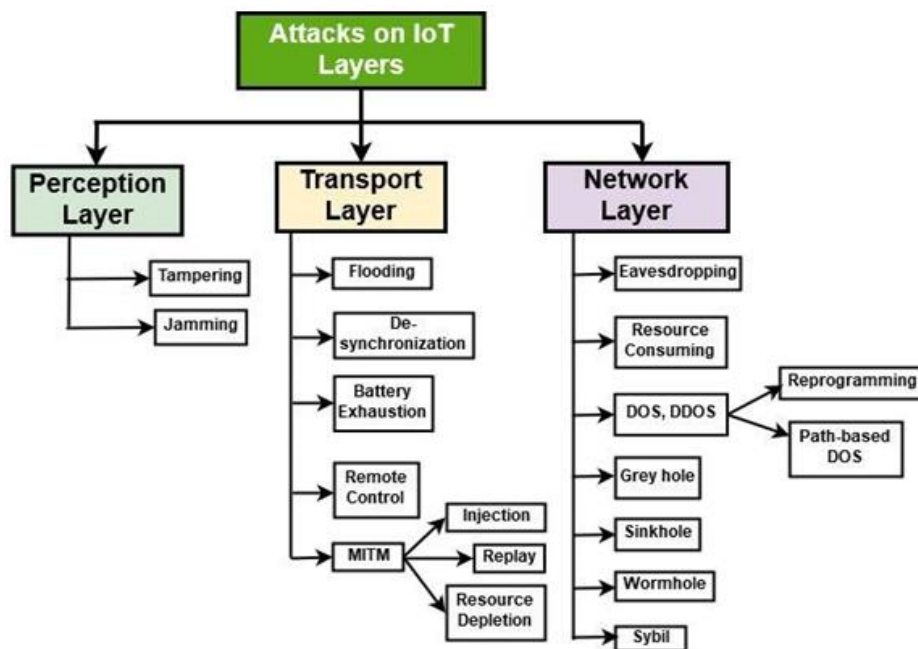


Figure 2. Attacks against IoT layers.

The types of attacks mentioned in Figure 2 are the ones, which are discussed by authors; there are other attacks, which have not been taken into consideration due non-popularity of the attacks or due to out of the scope of security mechanisms that are needed for such attacks.

3.1. Perception Layer

This is the first layer, which consists of the physical sensors and actuators of the IoT devices to sense the environment and collect information. The widespread attack at this layer is jamming and tampering. In a jamming attack, an adversary disrupts the operation of the network by squeezing/jamming the communication using high radio frequency signals [9]. Sometimes, an adversary can attack any sensor node to block the complete network resulting in a Denial of Service (DoS) or Distributed Denial of Service (DDoS) attack. Nowadays, cybercriminals use intelligent techniques to launch jamming attacks to evade various defensive measures like IDS/ Intrusion Prevention System (IPS). In defence to such attacks, a monitoring system is proposed by Liu et al. to distinguish interference and a real transmission where the energy consumed is verified each time to make sure it is not an attack [10]. This feature and energy monitoring system can identify channel interference efficiently but fall short for other attacks. Another model proposed using Monitor-Analyze-Plan-Execute (MAPE), which analysed signal strength but had similar drawback as of the previous one [11]. Multi-Agent Reinforcement Learning (MARL) algorithm is incorporated using Q-Learning to deal with jamming attacks and it gave 73% of performance. Likewise, an advanced Deep Learning (DL) framework is developed by researchers [12] to launch and mitigate jamming attacks. In this work, the Jammer senses the spectrum and if its classifier predicts any transmission to be successful, then the jammer blocks the transmission. Whereas, the defender system misleads the jammer decisions by propagating error signals. However, the success ratio of this model was very less & the maximum success ratio it gave was 69%. In both of these models, improved performance is indeed required to deal with real-time jamming attacks. From the discussion, we deduce that intelligent monitoring and learning models have the potential for detection of jamming attacks. Table 1 summarizes the discussed methods for a quick recap of the discussion.

Table 1. Perception layer attacks.

Attack	Technique & Implications	Defence Mechanism
WIRELESS JAMMING ATTACK	Jamming communication using high radio frequency signals [9].	Distributing the usage across the spectrum and continuous monitoring of cognitive spectrum [11].
		Accessing received signal strength by using MAPE architecture [12].
	Random and sensing-based jamming attacks using Deep Learning [12].	Deep Learning framework to divert and corrupt the jammer decisions [12].
	Complete jam of Wi-Fi signals and degradation of network performance	Reinforcement Learning for mitigating jamming based on Q-learning algorithm [13].

3.2. Transport Layer

This layer controls end-to-end links; and it mainly faces two types of attacks, flooding and the de-synchronization attacks. In flooding attack (TCP Synchronization / TCP-SYN), the memory resources of the devices are drained by propagating a control signal repetitively. Whereas, in the

de-synchronized attack, the attacker interrupts a fully established communication link between two genuine end nodes by re-synchronization (infinite cycle) of their transmission. It disrupts the communication and exhaust resources of the network. Such type of attacks leads to altering and draining out the network performance. A mitigation system based on rate-limiting model in Contiki Operating System (OS) proves efficient to identify UDP Flood attacks [14] but fails to work well in TCP. Early detection modules of Flooding attacks are developed by using Software Defined Networking (SDN) but the model lacked practical testing in real-time scenarios [15] [16]. Table 2 elucidates other cyber-attacks against this layer along with the security measures.

- 1) **Battery Exhaustion Attack:** It occurs due to more consumption of power while processing the tasks such as transmitting, maintaining and receiving data. An attacker injects malicious processing codes to elongate the task, sometimes making the device ineffective. This attack is most popular in mobile devices. An IDS is proposed by Nash et.al [17] to overcome this attack. The system monitors the battery level of the device and it estimates the power requirements for each task. When the power consumed is greater than the threshold estimated, it triggers an alert terminates the task to avoid exhaustion of the battery. However, IDS designed on one / two features is not able to unmask other attacks and requires customization as per the attack.
- 2) **Remote Control Attack:** In this attack, the attacker tries to intercept communication between two parties by using botnets or Man-in-the-Middle (MITM) attacks to gain full control of the device [18]. In some cases, the attacker may launch a DoS attack to disrupt resources or the whole device. Such kind of attack may cause devastating implications in wearable sensors or Medical IoT devices. Researchers have developed approaches to protect against Remote control attack by incorporating Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) for Constrained Application Protocol (CoAP) based LLNs [19]. CoAP is the widely used protocol in LLNs. Nevertheless, nowadays, many other protocols emerged for IoT networking such as MQ Telemetry Transport (MQTT), which consumes lesser energy of the devices. Hence, an approach has to be able to adapt to different protocols. Two other approaches shown in Table 2 are authentication and access control based, which are effective for Control Systems, Smart Grid, Home Automation and centralized control systems. However, is not effective for decentralized systems.
- 3) **Man in the Middle (MITM) Attack:** Weak security measures has given a stringent way for attackers to hold and vanish the resources of sensor devices. The unencrypted communication path is prone to attacks. An attacker can manipulate or delete information, violating the integrity, which may lead to various attacks: DoS, Eavesdropping, unauthorized access for tampering the data, injecting false information (authenticity) and Replay, Resource depletion and Injection attack [20]. In Eavesdropping, an adversary listens to the communication between the devices to know the capability and settings of the device to launch an attack. In a MITM attack, an attacker taps between two communicating devices by establishing a communication link and assuring them as authorized one by sending information to both and disconnecting their original communication link. It allows the intruder to acquire the user's data in an unethical way. The effective solutions for such attacks involve authentication and IDS system using Machine Learning, which give acceptable accuracy to defend against those attacks [21] [22].

Table 2. Transport layer attacks.

Attack	Technique & Implications	Defence Mechanism
Flooding Attack, ICMP/ TCP/ UDP/ HTTP/ DNS	A repetitively propagating control signal drains memory and battery [9]. May lead to DDoS attack or jamming attack.	Rate-limiting mechanism in Contiki OS [14].
		SDN based IDS for monitoring activity [16].
		Dynamic Anomaly Detection module by learning attack behaviour [15].
Battery Exhaustion Attack	Malicious codes to elongate the tasks & consumes more power, sometimes makes the device ineffective [17].	IDS monitors the power consumed for tasks. If greater than the threshold estimated, an alert is triggered [17].
Remote Control Attack	Intercepts communication using botnets or MITM attacks [18] to gain full control of the device and to disrupt resources of the whole device. Devastating in smart home, ICS, smart grid, power and energy management systems.	TLS & DTLS security model for CoAP based LLNs [19].
		Identity monitoring system for ICS using cryptography, image processing, authentication and authorization [23].
		Multi-path onion IoT gateways, hidden IoT nodes using Tor services making them accessible to only authorized users [24].
Man In The Middle (MITM) Attack	Attacker taps to manipulate or delete information. It can lead to DoS, replay, resource depletion and injection attack [20].	Supervised IDS for attack classification [21].
		Client-server model, Authenticating server's key with sensor data value [22].

3.3. Network Layer

This layer uses various technologies such as Radio Frequency Identification (RFID), Instrument flight rules (IFR), 3G, GSM, BLE, Universal Mobile Telecommunications System (UMTS), WiFi, ZigBee, etc for communicating with the devices. Communication in IoT devices occurs by routing and it is prone to various attacks [25]. Routing attacks involves spoofing, selective forwarding, altering routing paths or replaying packets, sinkhole, warm-hole etc. These attacks may lead to DoS threats. Table 3 shows some of the attacks against network or data link layer and its protective measures. While DoS or DDoS can be launched in Transport Layer also.

- 1) Eavesdropping: During transmitting the data from the sensor node to the gateway or server, the data is susceptible to hijack. An adversary can listen to the data and alter it from wireless channel [26]. An attacker detects information of the user and perceives the message-ID, timestamps; source and destination address which leads to a serious threat to privacy. Many solutions exist though, the latest framework for Eavesdrop resistance using Visible Light Communication (VLC) is a promising solution for IoT devices security [27].

- 2) **Resource consuming attacks:** Various attacks such as unfairness, collision and exhaustion attacks are included in this category. In an unfairness attack, the attacker tries to use whole services and resources of the application without considering the prerequisite it has [9]. Sometimes, this affects the network performance at the MAC layer. In a Collision attack, an attacker sends packets at the same frequency concurrently, which leads to collision and degradation of network performance. It manipulates frame header such that the checksum mismatch occurs, which leads to discarding of the data frames at the destination end. Exhaustion attack occurs when a channel is continuously active for long time to drain the battery power [9]. This kind of attacks lead to the failure in providing service and functionalities to end-users. These attacks can be mitigated using similar solutions of Battery Exhaustion and Flooding attacks in Transport Layer.
- 3) **Grey-hole attack:** In a multi-hop environment, the data transmission occurs from one node to another node in multiple steps [28]. In this process, the node forwards packets in the next hop to the destination (gateway). Before forwarding the packets, the attacker may misguide the route or inject malicious code to broadcast it further and initiate a routing loop. Such an activity is a Grey-hole attack in which the packets may loop infinitely deteriorating the performance of the network. The security mechanisms for such attacks [29] [30] [31] are explained in below subsections.
- 4) **Sinkhole attack:** In this type, a malicious node enchants with the neighbour nodes to create routes via malicious code. Once the attacker compromises the system, this attack creates an open door for other attacks [28]. It is very difficult to detect the sinkhole, selective forwarding and eavesdropping attacks in a network. Similar to this, in Sybil attack, a falsify node is present in the network with multiple fake identities deceiving the neighbouring nodes. Pretence, Masquerade and Replay attack mean the same. This attack also takes place in healthcare IoT devices, an illegitimate node behaves as a genuine node in the network, and it sends fake information to the remote area requesting treatment and an emergency team will respond to the non-existent patient [32]. This keeps the emergency staff busy, delaying and unattended to the real patients. A Denial-of-Service attack can be easily achievable by masquerade node. The captured data of masquerade node cause replay threat to the real-time IoT device application. Raza et al. [31] proposed an intrusion detection system for 6LoWPAN protocol targeting network routing attacks, sinkhole and selective-forwarding. The proposed IDS was developed using Contiki OS for IoT devices. It was successful to expose attacks in some situations but was unsuitable to smart home IoT devices.
- 5) **Wormhole attack:** wormhole attack is of similar kind in which an adversary receives packets from one location and then forwards and releases it to other location through a tunnel (wormhole). It is nearly impossible to detect or stop these types of attacks in a network using built-in security measures. Pongle et.al proposed an Intrusion Detection System to detect wormhole attacks in an IoT environment [30]. Nevertheless, the method is incapable of uncovering undefined cyber-attacks.
- 6) **Denial-of-Service attacks:** Data and network availability is a major security goal for IoT device applications. Mostly, in healthcare systems, threats of Denial-of-service are devastating because the devices and network need to be active and running all the time to monitor patients and to perform critical tasks [33]. Denial-of-Service and Distributed Denial-of-Service can affect the data, network performance and reliability of the whole network. There are two types of DoS attacks:
 - a) **Reprogramming Attack:** It refers to changing or modifying the source code. The application becomes inaccessible and sometimes it enters an infinite loop making the

service/ resource unavailable to the requester. Robust authentication, strong access control mechanism and continuous monitoring is a recommendable solution for such attacks [34].

- b) Path-based DoS: Numerous replay packets or spuriously injected packets overwhelms the sensor node by long-distance end-to-end communication path [35].

Researchers suggest a defensive approach based on the maximum magnitude of each middle-ware layer to handle such type of DoS/ DDoS attacks. The system checks for the number of requests sent to be under the predicted threshold capacity and if it exceeds, it triggers an alert to the network administrator [36] and blocks the request. Moreover, recent IDS approaches using Machine Learning (ML) and SDN proved efficient in blocking many DoS attacks [37] [38].

Table 3. Network layer attacks.

Attack	Technique & Implications	Defence Mechanism
Eavesdrop	The attack hijacks data during transmission [26] such that an adversary can listen or alter the data.	Innovative visible light communication (VLC) method based on channel correlation and error estimation [27].
Resource Consuming Attacks	Unfairness, Collision and Exhaustion attacks [9]. Failure in providing services.	Symmetric encryption and layered security mechanism using TLS [39].
Modification-type attacks (Routing attacks)	Grey hole, Sinkhole, Black hole and Wormhole attacks. [28]. Built-in security measures like authentication and access control cannot mitigate or detect such attacks.	IDS for sinkhole and selective-forwarding attacks [31].
		IDS to detect wormhole attacks in an IoT environment [30].
		Specification-based approach for the RPL protocol monitors network intrusions and malicious behaviour [29].
Sybil attack	A node with false identity, DoS or replay threat [32].	Host-based IDS using SDN blocks the victim device. SAAS model [40].
Denial-of-Service Attack	DoS, DDoS, Denial of Sleep, SYN Flood, DNS Flood, Ping Flood, UDP Flood, and ICMP Broadcast [41].	SDN architecture to identify DDoS, worm propagation and port scan [37]. IDS coupled provide better security.
		Evasion attacks against ML IDS can be mitigated using Gradient-based approach [38].

It is difficult to mitigate the attacks discussed by traditional security measures and needs up-gradation. We infer that the usual countermeasures involving basic mechanisms are ineffective. In most of the cases strong authentication, access control and monitoring systems are effective in identifying, mitigating and halting cyber-attacks. In addition, IDS is capable to detect most of the types of attacks in Perception, Transport and Network layer. The below section is elaborates and summarizes the potential security practices extracted from the above discussion.

4. SECURITY MEASURES

Numerous connected IoTs gives various decentralized ways for attackers or malware to enter. The high-security measures create a bottleneck for adaptability and make the device complex and in turn invite new security concerns [42]. IoT demands different customization for different purposes. The security incorporation should ensure the adaptability of the device and must be scalable with the addition of more devices to the network. The enhancement of the following security practices is required in IoT devices to ensure better protection and to ensure the security properties namely authentication, access control, confidentiality, integrity, non-repudiation.

4.1. Robust Authentication Mechanism

IoT devices has the feature of password authentication for accessing its services. Weak or default passwords, botnets, Trojans stealing passwords, dictionary and brute-force attacks are a point of high concern against authentication [43]. Nowadays, security specialists recommend integration of two methods for stronger authentication.

- 1) **Biometric Authentication:** Replacement of authentication process from password-based authentication to biometric authentication guarantees higher security, as it is robust against usual password cracking attacks. It involves bio-features of the authorized users such as face recognition, fingerprinting, eye recognition etc. Ruhul Amin et al. [44] proposed a biometric authentication protocol for IoT devices operating in a distributed cloud-computing domain to overcome vulnerabilities of cloud multi-server. Of course, biometric authentication have some issues such as cost and complexity of the algorithms used but many solutions exists in the literature for such loopholes.
- 2) **Multi-factor Authentication (MFA):** It involves multi-step authentication process: 2-step or 3-step, which includes a combination of knowledge-based (passwords), ownership-based (card), bio-based (fingerprint) features. One-time authentication requires two or three features (credentials) of the user, such as PIN and OTP for confirming a bank transaction. Biometric authentication integrated with MFA guarantees robust authenticity [45] in the current security implementations. As authentication is the primary requirement in smart devices, robust authentication mechanisms are recommendable for better protection.

4.2. A Robust Access Control Mechanism

Access control and data protection on low power IoT devices have become the need for protection against expanding cyberattacks. According to the research, Biometric access control is most favourable in IoTs. It takes the biological attribute of the individual for verification and identification. In this process, it compares the activities of the individual with the stored patterns in the system. This mechanism is vital to avoid host/ Internal-based attacks. To prevent unethical approaches for medical devices, a biometric-based two-level secure access control model is developed [46]. In this, the model converts the iris image to iris code. The verification of iris code is done by using hamming distance. It stores the master key in the system, employs less computation, and has a very small overhead. However, it involves a higher cost for biometric processing. To overcome this problem, many researchers have proposed advanced methods that minimize the cost of deploying. One such is framework has been developed using physical unclonable functions (PUFs) and hardware obfuscation by Nima et. al [47]. This method protects against access control circumvention and does not require key storage. This suggests that biometric or any other robust access control mechanism with less complexity guarantee security of IoT devices.

4.3. Software-Defined Networking (SDN)

Software-defined networking is the trending network security management in various application areas like business, smart homes and e-health systems. Any computer network consists of switches and routers as the main components. The important functions of switches/ routers are control plane and data plane. Control plane is responsible for where to send the traffic, whereas data plane forwards the traffic to a specific destination. In conventional networking, data plane and control plane are coupled. In SDN architecture, the control plane is separated from the data plane. A software-based entity, called controller, remotely controls the tasks of control plane [49]. The data plane executes in the hardware and control plane in the software and resides in a logically centralized way. SDN is capable to monitor network traffic and detect malicious activities. It identifies and isolates the compromised nodes from the rest of the network. Giotis et. al [37] used flow statistics in SDN architectures to spot abnormalities by using various ways such as launching a DDoS, worm propagation and port scan. It was efficient to detect attacks and does not cause overhead to the controller, but was not able to diagnose other attacks. However, SDN accompanied by an Intrusion Detection System is potential to identify or diagnose newer attacks [50] as per the latest research.

4.4. Intrusion detection system

Intrusion detection systems (IDS) is a program or algorithm, which tries to recognize malicious activities in a network. It also attempts to detect when a computer is under attack or an intruder is trying to compromise it. Besides, it identifies if a legitimate user is trying to escalate privileges or attempting to access unauthorized data or services. IDS has become an essential element for protecting the ICT infrastructure [50]. Nowadays, every network has IDS or IPS to detect and mitigate cyberattacks. According to the deployment model and data analysis, IDS is categorized as Network-based, Host-based or Application-based. In some contexts, system-based and application-based are considered as the two cases of Host-based IDS. Moreover, based on the technique / method used, IDS is categorized as Signature-based, Anomaly-based and Specification-based [51] [52]. An IDS system must distinguish attacks accurately, quickly and efficiently with less false alarms. Any IDS which identifies attacks accurately but takes a long time for detection is not suitable for current IoT networks [8]. Hence, it has become imperative to investigate a method that is capable to detect emerging attacks with less false alarms, which can handle a huge amount of data and take decisions quickly for real-time attack detection. The Signature-based system detects attacks based on signatures and known attack patterns but it is difficult to unmask known attack deviations or unknown attacks [53]. In literature, most of the implementations of IDS are rule-based which are inefficient in detecting novel attacks [32]. However, if the attack signatures database is up to date by adding new attack signatures every time, then this method is effective. Similarly, specification-based involves defining of rules by the administrator. In both these cases, the problem is the burden on the administrator to adapt to the changing number of devices and attacks. Anomaly Detection System detects deviations from a predefined normal behaviour but creates many false alarms for legitimate behaviours also when the user profile is complex and unknown. It is challenging to keep the IDS database up-to-date because of the heterogeneous network and changing environments such as network topology, servers, and several connected devices, communication protocols and open ports. To overcome this problem, the researchers are focusing on adaptable methods like Artificial Intelligence, Machine Learning and Deep Learning techniques [54].

4.4.1. Machine Learning IDS

This subsection provides details about the recent machine learning based Intrusion Detection Systems in IoTs. Mehdi et al. [40] proposed a host-based intrusion detection and mitigation system using OpenFlow protocol for security of smart home network. The scheme monitors the devices in home network to investigate the malicious activities and blocks the intruder to use the victim device once an intrusion is detected. The users in a smart home lack expertise in using security mechanisms, so Software Defined Networking (SDN) is employed in this model, providing Security as a Service (SaaS), such that a third party security specialist can monitor and take necessary actions when required. To avoid overburdening of nodes and communication network, host-based approach using filters is recommendable to monitor only suspicious nodes or malicious activities. In addition, the framework has the scalability to support heterogeneous new devices and technologies. The module consists of a database, which includes all the devices present in the smart home, their associated risks, and types of attacks and associated mitigation procedures for those. This model is based on Machine learning techniques which uses learned signature patterns of known attacks. For this process, a sensor element gathers data traffic from suspicious nodes and send it to SDN controller. The captured traffic is transformed to service provider for feature extraction and to create predictive models of attacks. IoT Intrusion Detection and Mitigation (IDM) model uses linear regression and Software Vector Machine (SVM) to create a classification model, based on which attack is identified. Once an attack is detected, an alarm is raised, the victim node and attacker are identified and/or mitigation is done if measures are available in the database. This model was tested on a real IoT device, a smart lighting system and was proved efficient to detect the attack. The major disadvantage of this approach is that, each time a new device is added to the network, it has to be manually updated in the database. Only specific devices can be monitored and this approach is not feasible to investigate all devices in a home network. Moreover, in current zero-day attacks scenario, this approach is unsuitable as unknown attacks are not detected which may have devastating implications on the end-users.

Similarly, Kleber et al. [55] proposed a mechanism to overcome the huge number of IDS alerts, which are triggered in a conventional IDS system. This model is based on the fusion of various events, security logs and alerts and is not concerned with network traffic. The proposed scheme gathers raw data and change it in a standard normalized format. Then these normalized events are clustered into meta-events, to represent possible attack scheme more clearly when compared to the disconnected alerts. With this situational awareness, in the final state, the meta- events are classified using machine learning to categorize it as an attack or false alarm. SVM, Decision Tree and Bayesian Network have validated the classification scheme. This was tested using DARPA Intrusion Evaluation challenge [56] and SotM from the honeynet and the accuracy was in between 40 - 60% in attack detection with lesser false positive rates and was able to detect some of the newer attacks as well. However, this model was not been tested for current zero-day attacks and may not be feasible to detect multi-stage attacks. Various improvements are necessary in terms of security and complexity of the classification taxonomy of the approach.

Heena et al. [57] developed another machine learning approach for wireless sensor network security based on human immune system. This method intelligently detect anomalies by classifying the nodes into two categories: fraudulent or benevolent nodes. After which, the mechanism create virtual antibodies and depending on that, the gateway takes a decision whether or not to attack the fraudulent nodes. The model works similar to human immune system as a second line defense in the body. However, the actual implementation of the proposed mechanism was not provided. Likewise, Sara et al. [58] proposed an IDS Machine Learning based on feature selection and clustering algorithm incorporating filter and wrapper methods using linear correlation coefficient (FGLCC) algorithm, cuttlefish algorithm (CFA) and Decision Trees for

classification. The authors verified the proposed method using KDD Cup 99 large data sets, which gave 95% detection rate.

4.4.2. Deep Learning IDS

Feature extraction and data classification has emerged as efficient techniques for IDS. However, most of the proposed approaches are inefficient when dataset is of large size. Kabir et al. proposed an Intrusion Detection System using Least Square Support Vector Machine (LS-SVM) in which the attack detection is done in 2 steps. In the first step, the entire dataset splits into subgroups such that they represent the whole dataset. In the second step, LS-SVM is applied to the proposed algorithm to determine intrusions. Various experiments using KDD 99 database proved it an efficient algorithm for intrusion detection [59]. The advantage of this method is that it supports static and incremental data also. Papamartzivanos et al. [50] proposed an intelligent adaptive misuse Intrusion Detection System using Deep Learning. This method can adapt and sustain to various network environments with higher rates of attack detection. They used autonomic computing Self-Taught Learning method supported by MAPE-K model to assist IDS in new environments. This model is integrated with MAPE-K method to create a framework for the autonomous and adaptive system. The model has been evaluated with various environmental changes and was capable to adapt with detection rate of approximately 73.3 % by not only detecting the attack but also categorizing it so that solution can be found easily. The benefits of deep learning methodologies is training the IDS based on the network activity in new environment.

From the above discussion, we deduce few quantitative metrics in Table 4, for an IDS system to determine its effectiveness.

Table 4. IDS Quantitative Metrics

Metrics	Description
Coverage	The number and types of attacks that IDS can detect in a realistic environment.
Handling Traffic Bandwidth	The Ability of the IDS to handle High bandwidth traffic, block or resist traffic greater than the bandwidth of the channel.
Resisting attacks against IDS	Few attackers target IDS so that when the IDS is compromised, it becomes easier to attack the network and devices. Therefore, IDS must be capable to withstand attacks targeted against it.
Probability of Detection	It defines how accurately the system detects an intrusion. The approaches discussed have shown the accuracy up to 70 - 90%. Nevertheless, real-time networking demands higher accuracy in the detection of various attacks.
Probability of False Alarms [8]	Sometimes a non-attack activity is categorized as attack, and vice versa. These types of decisions of IDS may cause fatal implications. Latest Machine Learning and Deep Learning algorithms use Confusion matrix to correctly classify an event.
Ability to Detect Unknown Attacks [8]	The concern for Zero-day attacks are growing day by day. A system that is not able to detect unfamiliar attacks is inefficient.
Ability to Identify an Attack	How correctly a system is identifies an attack is important to take further actions by the network administrators. If the attack is categorized wrongly, for example, Wormhole attack is categorized as Grey hole; it may mislead the administrator by the risk level of the attack.
Ability to Determine Attack Success	The system must also be able to determine the status of the attack, such as, its success or failure, to what extent it is successful, and to what extent it damaged the resources.
Others	Other measurements include ease of use, deployment and maintenance. The IDS must meet resource requirements, performance and quality of service.

4.5. Light-Weight Encryption

To secure the data at rest & data in transfer and to maintain confidentiality & integrity, encryption is necessary, which encapsulates the data in an unreadable format and it is decrypted at the receiver. Most common types of encryption algorithms are symmetric and as symmetric. Each of these approaches is capable to protect data against some attacks that another approach find it difficult. However, as-symmetric algorithms involve more processing which is unsuitable for IoT devices [60]. Currently, Physically Unclonable Function (PUF) is utilized for adding extra security hardware layer to protect against perception or physical layer attacks [61] and for end-points security. This method consumes lesser resources compared to other encryption algorithms. Whereas for communication security symmetric-key encryption is suitable, one such lightweight symmetric key encryption scheme has been developed [62], which provided very effective in securely transferring data in IoT networks.

The collaboration of the above-mentioned security properties is capable to defend against a maximum number of cyberattacks. In the below section, we provide an appropriate way of incorporating these security mechanisms in IoT architecture to acquire utmost protection.

5. SECURE IOT FRAMEWORK

From the above discussion, we deduced that few security mechanisms are significant to provide robust security for IoT devices. The summary of the findings from the study is shown in Figure 3, which is a proposal of secure architecture for IoT devices to protect it from malicious threats.

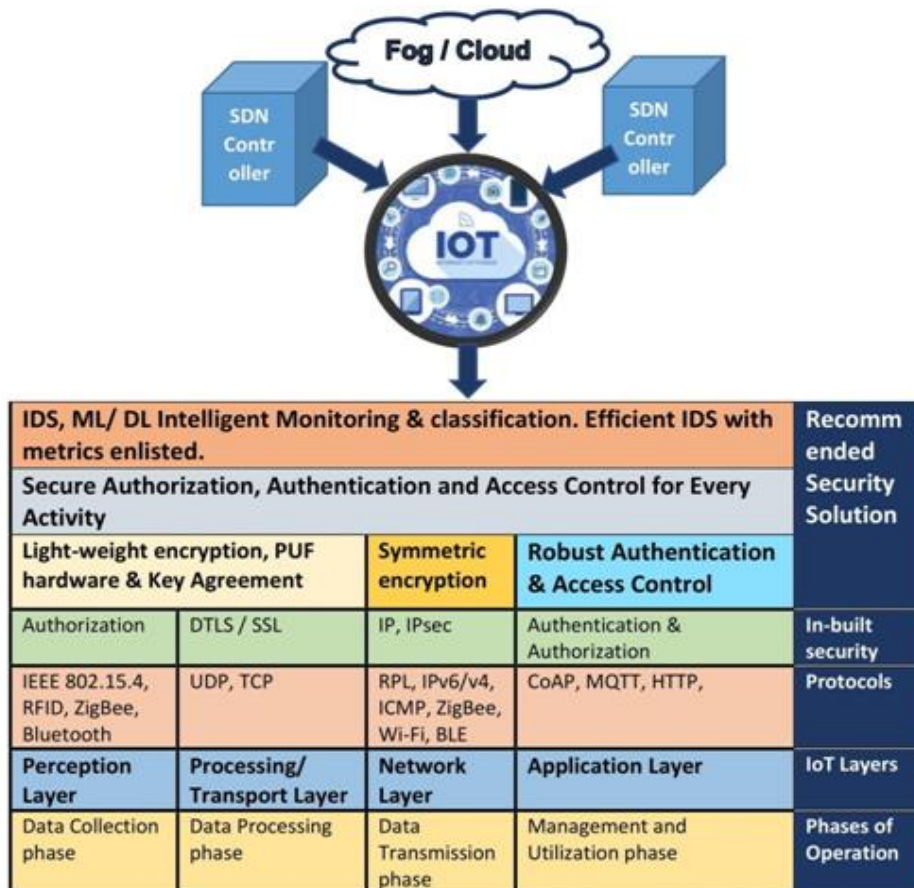


Figure 3. Secure IoT framework

While manufacturing the IoT devices, a standardized security layer needs to be included to provide basic security. The figure is explained from bottom Phases of operation to the topmost recommended security solution. The phases of operation and Layers of IoTs are already discussed in section 2. CoAP, MQTT and HTTP are frequently used protocols in Application Layer. Transport Layer consists of TCP and UDP protocols; similarly, the protocols of the other two layers are mentioned. Each layer has low-level in-built security, such as, to access application layer utilities the user has to authenticate his identity. At the network layer Internet Protocol Security (IPsec) is incorporated for secure communication. Similarly, at the transport layer, DTLS & Secure Software Layer (SSL) provides security and the perception layer customizes its activities based on the authorization process. The fundamental security controls in IoT architecture lag behind to defend it against current cyberattacks.

According to this research, we have provided the necessary security mechanism at each layer to ensure optimal protection against cyberattacks. The security mechanisms discussed are placed appropriately to suit the layer requirements in the IoT architecture.

- 1) Robust authentication and access control: The application layer attacks can be mitigated using robust authentication and access control mechanisms provided.
- 2) Symmetric Encryption & Light-weight cryptography: Transport and Network layer needs lightweight encryption combining the features of both symmetric and as- symmetric encryption to secure the ends and the data in transmission.
- 3) Secure authorization, authentication and access control: Every action at different layer needs to be checked for its authorization with proper access control.
- 4) Intelligent IDS: For the transport and network layer, regular monitoring can reduce the number of malicious intrusions. The attacks at the perception layer also can be mitigated by IDS monitoring and key agreement protocols.
- 5) Software Defined Networking: These days SDN architecture provides better security compared to other networking practices. Due to programming capability of SDN, secure and easily controllable network can be designed.

5.1. Evaluation of Proposed Framework

Putchala et al. [63] proposed a distributed multi-layered IDS architecture for IoT devices to ensure identifying malicious intrusions at each layer efficiently and accurately. The author suggested placement of Deep Learning based IDS at each layer for maximum coverage and better complexity. The implementation is tested and proved efficient. This suggest that IDS placed at all layers of IoT architecture guarantees better attack detection compared to one specific layer based IDS. Therefore, recommended security solution, IDS at all layers proved efficient. Lightweight encryption and authentication protocol proposed by researchers show that the protocol is protected against possible security threats [44]. In similar way, a robust cryptographic technique can be incorporated and tested along with the IDS system. The other mechanisms can be tested solely or in combination of all mechanisms to validate its effectiveness. In future, we aim to test all the mechanisms suggested and prove its effectiveness in terms of complexity, resource constraint requirements and various features.

The following are the metrics based on which the proposed framework has to be evaluated.

- 1) Processing response time: Performance of IoT device in a real environment after implementing all the recommended security mechanisms.

- 2) Resource consumption: The level of resources consumed, such as battery power, processing power and memory used.
- 3) Attacks mitigation: The number of attacks that are mitigated after implementing the framework, the accuracy of attack detection.
- 4) Scalability: The amount of data that is easily being processing without overwhelming device.

Other metrics are also included in future during the implementation of the framework and compare its effectiveness with the recently proposed secure IoT architectures. The framework needs to be tested by launching real-time attacks against the IoT device to ensure its deployment in current IoT devices.

6. CONCLUSION

With the advancement of heterogeneous smart IoT devices, the concern for security is increased. In this paper, we have provided various types of attacks against IoTs and their protective measures. Certainly, many other attacks are launched against IoT layers, but the attacks discussed here are recurrent and devastating. We have learnt that securing the endpoints, network monitoring and the protecting data in the transfer is mandatory, to detect and prevent malicious activities in IoTs. Thus, we have proposed a framework incorporating Robust Authentication, Robust Access Control, Lightweight cryptography and Intrusion detection system, to secure data in transfer, sensitive stored data, settings & privileges. SDN is a trendy networking paradigm to securely control the whole network using programming. Machine Learning and Artificial Intelligence (AI) is an emerging field for IDS, which allows a system to learn, deduce and decide based on cognitive functions of pattern recognition and computational learning theory without any programming. In addition, the metrics used by IDS to identify different attacks against IoT layers are provided. The proposed framework is a valuable contribution to the IoT architecture due to its holistic approach of combination of various potential security mechanism. In future, this research aims to implement the framework and validate its effectiveness in terms of security, performance and usability.

REFERENCES

- [1] Koliass, Constantinos, Angelos Stavrou, and Jeffrey Voas. "Securely Making" Things" Right." *Computer* 48, no. 9 (2015): 84-88.
- [2] Hilton, Scott, Dyn analysis summary of Friday, October 21 attack (2016), <https://dyn.Com/blog/Dyn-analysis-summary-of-friday-october-21-attack>
- [3] Omanović-Miklićanin, E., Maksimović, M., & Vujović, V. (2015). The future of healthcare: nanomedicine and internet of nano things. *Folia Medica Facultatis Medicinae Universitatis Saraeviensis*, 50.
- [4] Forecast economic impact of the Internet of Things (IoT) in 2025 (in billion U.S. dollars).
- [5] Borgia, Eleonora. "The Internet of Things vision: Key features, applications and open issues." *Computer Communications* 54 (2014): 1-31.
- [6] Santos, Leonel, Carlos Rabadao, and Ramiro Gonçalves. "Intrusion detection systems in Internet of Things: A literature review." 2018 13th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2018.
- [7] McDermott, Christopher D., Farzan Majdani, and Andrei V. Petrovski. "Botnet detection in the internet of things using deep learning approaches." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.
- [8] Tabassum, Aliya, Aiman Erbad, and Mohsen Guizani. "A Survey on Recent Approaches in Intrusion Detection System in IoTs." 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, 2019.

- [9] Alaba FA, Othman M, Hashem IA, Alotaibi F. Internet of Things security: A survey. *Journal of Network and Computer Applications*. 2017 Jun 15; 88:10-28.
- [10] Liu, Wei, et al. "Various detection techniques and platforms for monitoring interference condition in a wireless testbed." *Measurement methodology and tools*. Springer, Berlin, Heidelberg, 2013. 43-60.
- [11] Ashraf, Qazi Mamoon, and Mohamed Hadi Habaebi. "Autonomic schemes for threat mitigation in Internet of Things." *Journal of Network and Computer Applications* 49 (2015): 112-127.
- [12] Erpek, Tugba, Yalin E. Sagduyu, and Yi Shi. "Deep learning for launching and mitigating wireless jamming attacks." *IEEE Transactions on Cognitive Communications and Networking* 5.1 (2018): 2-14.
- [13] Aref, Mohamed A., Sudharman K. Jayaweera, and Stephen Machuzak. "Multi-agent reinforcement learning-based cognitive anti-jamming." *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017.
- [14] Malik, Manisha, and Maitreyee Dutta. "Contiki-based mitigation of UDP flooding attacks in the Internet of things." *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017.
- [15] Bhunia, Suman Sankar, and Mohan Gurusamy. "Dynamic attack detection and mitigation in IoT using SDN." *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2017.
- [16] Wani, Azka, and S. Revathi. "Analyzing threats of iot networks using sdn based IDS (sdnet-ids)." *International Conference on Next Generation Computing Technologies*. 2017 Springer, Singapore, 2017.
- [17] Nash DC, Martin TL, Ha DS, Hsiao MS. Towards an Intrusion detection system for battery exhaustion attacks on mobile computing devices. In *Third IEEE international conference on pervasive computing and communications workshops* 2005 Mar 8 (pp. 141-145). IEEE.
- [18] Arias O, Wurm J, Hoang K, Jin Y. Privacy and security in internet of things and wearable devices. *IEEE Transactions on Multi-Scale Computing Systems*. 2015 Apr 1; 1(2):99-109.
- [19] Brachmann, Martina, et al. "End-to-end transport security in the IP-based internet of things." *2012 21st International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2012.
- [20] Khan MA, Salah K. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*. 2018 May 1; 82:395-411.
- [21] Anthi, Eirini, et al. "A Supervised Intrusion Detection System for Smart Home IoT Devices." *IEEE Internet of Things Journal* (2019).
- [22] Kara, Mustafa, and M. U. R. A. T. Furat. "Client-Server Based Authentication against MITM Attack via Fast Communication for IIoT Devices." *Balkan Journal of Electrical and Computer Engineering*, 2018, 6.2 (2018): 88-93.
- [23] Condry, Michael W., and Catherine Blackadar Nelson. "Using smart edge IoT devices for safer, rapid response with industry IoT control operations." *Proceedings of the IEEE* 104.5 (2016): 938-946.
- [24] Yang, Lei, et al. "Hide your hackable smart home from remote attacks: The multipath onion IoT Gateways." *European Symposium on Research in Computer Security*. Springer, Cham, 2018.
- [25] Madakam S, Ramaswamy R, Tripathi S. Internet of Things (IoT): A literature review. *Journal of Computer and Communications*. 2015, May 25; 3(05):164.
- [26] Mahmoud R, Yousuf T, Aloul F, Zualkernan I. Internet of things (IoT) security: current status, challenges and prospective measures. *10 th International Conference for Internet Technology and Secured Transactions (ICITST) 2015*, Dec 14 (pp. 336-341). IEEE.
- [27] Liu, Xiangyu, et al. "SecLight: A New and Practical VLC Eavesdropping-Resilient Framework for IoT Devices." *2019, IEEE Access* 7 (2019): 19109-19124.
- [28] Airehrour D, Gutierrez J, Ray SK. Secure routing for internet of things: A survey. *Journal of Network and Computer Applications*. 2016 May 1; 66:198-213.
- [29] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based ids fordetecting attacks on rpl- based network topology," *Information*, 2016, vol. 7, no. 2, p. 25, 2016.
- [30] Pongle P, Chavan G. Real time intrusion and wormhole attack detection in IoTs. *International Journal of Computer Applications*. 2015 Jan 1; 121(9).
- [31] Raza S, Wallgren L, Voigt T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks*. 2013, Nov 1; 11(8):2661-74.
- [32] Le A, Loo J, Lasebae A, Aiash M, Luo Y. 6lowpan: a study on QoS security threats and countermeasures using intrusion detection system approach. *International Journal of Communication Systems*. 2012 Sep; 25(9):1189-212.

- [33] Rathore H, Mohamed A, Al-Ali A, Du X, Guizani M. A review of security challenges, attacks and resolutions for wireless medical devices. 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC) 2017 Jun 26 (pp. 1495-1501). IEEE.
- [34] Sonar, Krushang, and Hardik Upadhyay. "A survey: DDoS attack on Internet of Things." *International Journal of Engineering Research and Development* 10, no. 11 (2014): 58-63.
- [35] Deng, Jing, Richard Han, and Shivakant Mishra. "Defending against path-based DoS attacks in wireless sensor networks." In *Proceedings of the third ACM workshop on Security of ad hoc and sensor networks*. pp. 89-96. ACM, 2005.
- [36] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE wireless communications*, 2004 vol. 11, no. 1, pp. 48– 60, 2004.
- [37] Giotis, Kostas, Christos Argyropoulos, Georgios Androulidakis, Dimitrios Kalogeras, and Vasilis Maglaris. "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments." *Computer Networks* 62 (2014): 122-136.
- [38] Biggio, Battista, Iginio Corona, Davide Maiorca, Blaine Nelson, Nedim Šrnđić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. "Evasion attacks against machine learning at test time." 2013 In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387-402. Springer, Berlin, Heidelberg, 2013.
- [39] King, James, and Ali Ismail Awad. "A distributed security mechanism for resource-constrained IoT devices." *Informatica* 40.1 (2016).
- [40] Nobakht, Mehdi, Vijay Sivaraman, and Roksana Boreli, A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow, 11th International conference on availability, reliability and security (ARES). IEEE2016
- [41] Montoya, Maxime, et al. "SWARD: A Secure Wake-up RaDio Against Denial-of-Service on IoT Devices." *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2018.
- [42] Suo, Hui, Jiafu Wan, Caifeng Zou, and Jianqi Liu. "Security in the internet of things: a review." In 2012 international conference on computer science and electronics engineering, vol. 3, pp. 648-651. IEEE, 2012.
- [43] Hossain, M. Shamim, Ghulam Muhammad, Sk Md Mizanur Rahman, Wadood Abdul, Abdulhameed Alelaiwi, and Atif Alamri. "Toward end-to-end biometric-based security for IoT infrastructure." *IEEE Wireless Communications* 23, no. 5 (2016): 44-51.
- [44] Amin, Ruhul, et al. "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment." *Future Generation Computer Systems* 78 (2018): 1005-1019.
- [45] Ometov, Aleksandr, et al. "Challenges of Multi-Factor Authentication for Securing Advanced IoT Applications." *IEEE Network* 33.2 (2019): 82-88.
- [46] Ali, Bako, and Ali Awad. "Cyber and physical security vulnerability assessment for IoT-based smart homes." *Sensors* 18, no. 3 (2018): 817.
- [47] Karimian, Nima, et al. "Secure and Reliable Biometric Access Control for Resource-Constrained Systems and IoT." *arXiv preprint arXiv:1803.09710* (2018).
- [48] Catarinucci, Luca and De Donno, Danilo and Mainetti, Luca and Palano, Luca and Patrono, Luigi and Stefanizzi, Maria Laura and Tarricone, Luciano, An IoT-aware architecture for smart healthcare systems, *IEEE Internet of Things Journal*, {515--526}, 2015.
- [49] McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. "OpenFlow: enabling innovation in campus networks." *ACM SIGCOMM Computer Communication Review* 38, no. 2 (2008): 69-74.
- [50] Papamartzivanos, Dimitrios and Marmol, Felix Gomez and Kambourakis, Georgios, *Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems*, 2019.
- [51] Mell, Peter, *Understanding intrusion detection systems*, *IS Management Handbook*, 409--418, 2003, Auerbach Publications
- [52] Morin, Benjamin and Me, Ludovic and Debar, Herve and Ducasse, Mireille, M2D2: A formal data model for IDS alert correlation, *International Workshop on Recent Advances in Intrusion Detection*, 115--137, 2002
- [53] Pacheco, Jesus, and Salim Hariri. "IoT security framework for smart cyber infrastructures." 2016 *IEEE 1st International Workshops on Foundations and Applications of Self Systems (FAS W)*. IEEE, 2016.

- [54] Fadlullah, Zubair Md, et al. "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems." *IEEE Communications Surveys & Tutorials* 19.4 (2017): 2432-2455.
- [55] Stroeh, Kleber, Edmundo Roberto Mauro Madeira, and Siome Klein Goldenstein, An approach to the correlation of security events based on machine learning techniques, *Journal of Internet Services and Applications* 4.1, 2013
- [56] The 1999 DARPA off-line intrusion detection evaluation, Lippmann, Richard and Haines, Joshua W and Fried, David J and Korba, Jonathan and Das, Kumar, *Computer networks*, 579-- 595, 2000
- [57] Rathore, Heena, and Sushmita Jha, Bio-inspired machine learning based wireless sensor network security, 2013 World Congress on Nature and Biologically Inspired Computing, IEEE, 2013
- [58] Mohammadi, Sara and Mirvaziri, Hamid and Ghazizadeh-Ahsaei, Mostafa and Karimipour, Hadis, Cyber intrusion detection by combined feature selection algorithm, author, *Journal of information security and applications*, 80-88, 2019
- [59] Kabir, Enamul and Hu, Jiankun and Wang, Hua and Zhuo, Guangping, A novel statistical technique for intrusion detection systems, *Future Generation Computer Systems*, 303-318, 2018
- [60] Stergiou, Christos, et al. "Secure integration of IoT and cloud computing." *Future Generation Computer Systems* 78 (2018): 964-975.
- [61] Bolotnyy, Leonid, and Gabriel Robins. "Physically unclonable function-based security and privacy in RFID systems." *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*. IEEE, 2007.
- [62] Rajesh, Sreeja, et al. "A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices." *Symmetry* 11.2 (2019): 293.
- [63] Putchala, Manoj Kumar. "Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)." (2017).

HYBRID APPLICATION LAYER PROTOCOL DESIGN FOR IOT ENVIRONMENTS

Erdal ÖZDOĞAN¹ and O.Ayhan ERDEM²

¹Department of Information Systems, Gazi University, Ankara, Turkey

²Department of Computer Engineering, Gazi University, Ankara, Turkey

ABSTRACT

One of the important factors affecting communication performance in the Internet of Things is the messaging protocol. MQTT, XMPP and AMQP are centralized application protocols that communicate through the server. DDS and CoAP are application protocols that can communicate directly, especially in real-time applications. As the Internet of Things is becoming more widespread and usage scenarios have different requirements, new approaches to data communication are required. In this study, a UDP based hybrid application layer protocol has been designed which can communicate both directly and through central server. In addition, operating logic and packet structure of the developed hybrid protocol is examined.

KEYWORDS

Internet of Things, IoT Application Protocol, Hybrid IoT Protocol, Lightweight IoT Protocol, UDP Based IoT protocol

1. INTRODUCTION

The unpredictable development of the Internet, from its inception to the present day, shows that developments and innovations in Internet technology will have great repercussions worldwide in the coming years. In 2012, the number of devices connected to the Internet exceeded the number of people living on earth, and by 2020, between 26 and 50 billion objects are expected to be connected to the Internet and are expected to produce millions of gigabytes of data [1]. Organizing, interpreting and transforming this enormous mass of data to be produced in a heterogeneous structure is seen as one of the most important problems that the Internet of Things (IoT) will face [2]. The idea that every device or object in the near future will have an IP address, either directly or indirectly, reveals the need to develop protocols that support IP [3] [4]. The inadequacy of the existing Internet protocols to meet this aim leads to the development of new protocols. For this purpose, in order to provide data transfer on IoT, Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Data Distribution Service (DDS), and Extensible Messaging and Presence Protocol (XMPP) have been developed or adapted to IoT requirements [5]. On the other hand, the report “M2M service layer: APIs and protocols overview”, published by ITU-T in 2014, states that new protocols should be developed in order to meet developing technologies and different usage scenarios [6].

MQTT, AMQP and XMPP protocols are central protocols that use servers for data transfer [7]. In this approach, there is no direct communication between the client (or user) and the IoT Device. Protocols such as CoAP and DDS are direct communication protocols that work in decentralized approach [8],[9]. Both centralized and the decentralized approaches have several advantages and disadvantages. One of the most important advantages of the decentralized approach is its ability to communicate directly [10]. However, administrative difficulties are the main disadvantages of this architecture [11]. One of the most important advantages of central communication is that the communication can be controlled by the server. However, the use of the server causes a single point of failure [8], [12] - [14] [15].

In this study, a hybrid application layer protocol (hIoT) is designed for data communication which provides both server-based and direct communication. The designed protocol has been developed in order to transfer low-dimensional data generated by sensors in devices with limited resources and in low bandwidth environments. In the developed protocol, both communication methods can be used for different purposes and needs. In addition, dynamic switching can be made between these two methods.

In the second part of the article, academic studies on IoT protocols are discussed. In the third section, the general working principle of the designed application protocol, packet structures, and protocol components are examined. In the last section of the article, a summary of the study and the advantages of the proposed protocol are given and suggestions are made to shed light on future studies.

2. RELATED WORK

Protocols play an important role in the realization of IoT. Research on existing IoT protocols shows that these protocols can be superior in different areas when compared with each other. In this section, studies on the performance comparisons in terms of bandwidth, latency, and interoperability of IoT application protocols are discussed.

In the study of Thangavel et al. [16], where the middleware layer was developed in order to enable MQTT and CoAP protocols to work together, MQTT and CoAP protocols were discussed in terms of end-to-end latency and bandwidth consumption. According to this study, MQTT is more successful in environments with packet losses of less than 20%; however, CoAP is more successful in higher packet losses. In a study [17] comparing protocols according to their payload, it was reported that the CoAP protocol experienced a loss of performance at payload capacities greater than 1024 bytes. In the study conducted in mobile and unstable networks [18], MQTT and AMQP protocols were compared in terms of bandwidth usage, delay, and jitter effect. There was no significant difference between these protocols. MQTT is more successful in terms of energy efficiency. In another study comparing CoAP and MQTT protocol in transporting the same data [19], CoAP protocol was stated to be more efficient. In a study conducted in an environment with high network traffic, the MQTT protocol was reported to be more successful than CoAP, with a higher bandwidth and lower latency [20].

In a study addressing the problem of service discovery in the IoT, M. Kirsche et al. stated that MQTT and CoAP protocols could not provide an end-to-end communication, delays occurred due to message conversion, and the discovery process was complex. They have therefore developed a simplified structure using the combination of mDNS and DNS-SD [21] [22].

In a study that can be examined under the title of Developing IoT Protocols [23], it is mentioned that MQTT and CoAP protocols are widely used in IoT, but both protocols face scalability problems in large networks with multiple sensors. In the study, in order to ensure scalability and

use of bandwidth efficiently, CoAP protocol was improved. Accordingly, it was ensured that the sensors were grouped to form a cluster and a representative sensor sent the data, which resulted in 18% less bandwidth consumption.

The complexity of the IoT system and the increase in the number and diversity of devices connected to the network lead to the use of hybrid methods as solutions. In the study of Bellavista et al., the study states that this architecture developed in high density networks provides scalability [8].

The study [24] conducted to ensure that multiple protocols work together using a central middleware layer reveals the interoperability of protocols without significant performance losses. In the design of an application protocol called “Custom UDP”, the proposed protocol was compared with various IoT protocols and according to experimental results, it is stated that it offers relatively low energy and bandwidth consumption in environments with unpredictable packet losses [25].

As it is seen, it is very difficult to state that only one protocol is superior in every aspect of the IoT. Different protocols can be used according to the network topology used, security need, scalability, and bandwidth usage. In addition, the diversity of middleware software makes it difficult to connect to IoT devices and interpret the collected data [26]. The heterogeneous nature of the IoT ecosystem brings along the need for different protocols with an ever increasing trend of growth.

3. DEVELOPED APPLICATION LAYER PROTOCOL

Within the scope of the IoT, various research and academic studies have been carried out in recent years on problem situations such as security, resource discovery, interoperability, compatibility, and performance improvements. This study does not focus on all of these problem situations and some assumptions are taken into consideration. Accordingly, although the concept of security is critical for IoT, this study assumes that the environment is safe and data security is out of scope. The location and service information of the sensor was taken into account in order to comply with MQTT and CoAP protocols during the registration and resource discovery process. Parameters such as sensor manufacturer and sensor type have not been taken into consideration. The location of sensor, and the service provided by the sensor is designed as shown in Figure 1 to provide compatibility with the “topic” used in the MQTT.

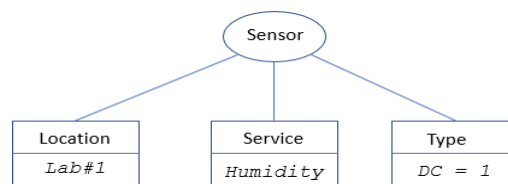


Figure 1. Sensor properties used in the developed protocol

In this study, it is assumed that the data obtained from the sensors can be carried in a single data packet and the IP packets are not fragmented.

Since the devices used in the IoT system have limited resources, applications requiring high processing power, memory and bandwidth consumption are inadequate and inefficient in meeting the need. Therefore, the developed protocol is designed to be flexible and simple to operate at low resources.

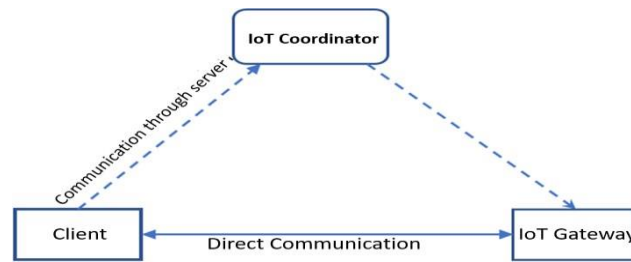


Figure 2. Hybrid Application Protocol supporting direct communication and server-based communication

Within the scope of the study, a Hybrid Application Layer Protocol (hIoT) was developed, as shown in Figure 2, which works with both the central server and supports direct access. Depending on the sensor type, either direct communication or communication through server can be selected. When the server is failed, a dynamic switching mechanism is designed that can be switched to direct communication. Thus, it is aimed to prevent a single point of failure in server-based communications.

3.1 General Operating Principle

In order for the designed system to function correctly, the roles of devices in the IoT ecosystem must be defined precisely.

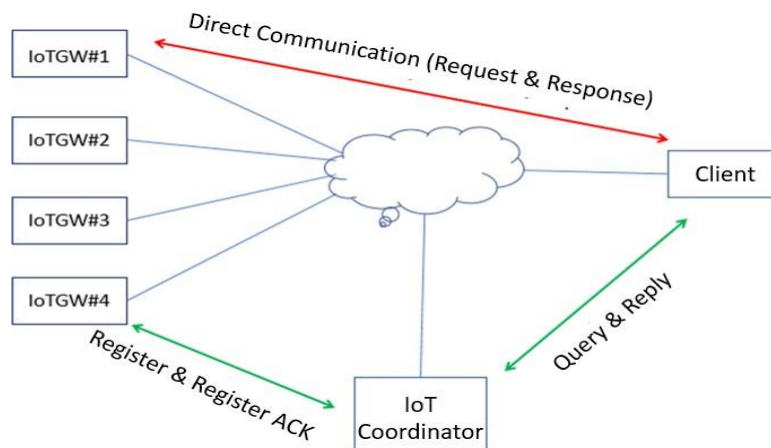


Figure 3. Devices and communication packets in the developed protocol

In the developed protocol, as shown in Figure 3, three different components are defined: IoT Coordinator, IoT Gateway (IoTGW), and Client.

The operation of the protocol is examined in three phases: Service Registration Phase, Service Query Phase, and Data Transfer Phase, as shown in Figure 4.

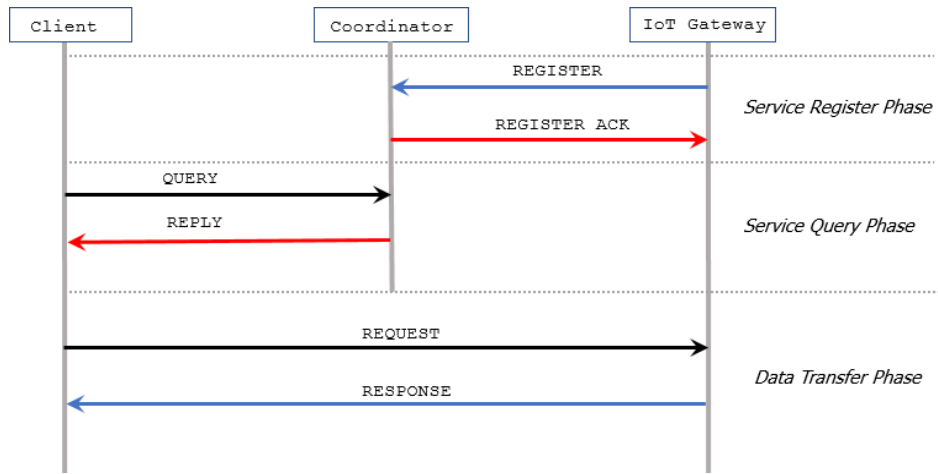


Figure 4. Basic phases of the developed protocol

The service register phase is the registration of the services running on the IoT Gateway to the database. During the service query phase, clients are asked how to access the services they are interested in. Information about how to access the service by the coordinator is sent to the client in response. In the data transfer phase, direct communication between the client and the gateway or communication through the server is performed, depending on the access information given by the coordinator.

3.2 Packet Types

The designed IoT protocol includes eight different packets. Packet types and their descriptions are shown in Table 1.

Table 1. Packet Types and Descriptions

Packet Type	Description
Control	Used for accessibility control.
Reset	Resets the access method.
Register	Enables the service provided by IoTGW to be registered to the server.
Error	The packet sent in case of error.
Query	The query packet that the client use makes for service discovery.
Reply	Sent in response to the service discovery packet.
Request	A request for information from the service provided by IoTGW.
Response	A packet containing data that is sent in response to the request packet.

3.3 Packet Structure and Headers

The header of the developed protocol is designed in a structure consisting of the areas shown in Figure 5 for ease of use.

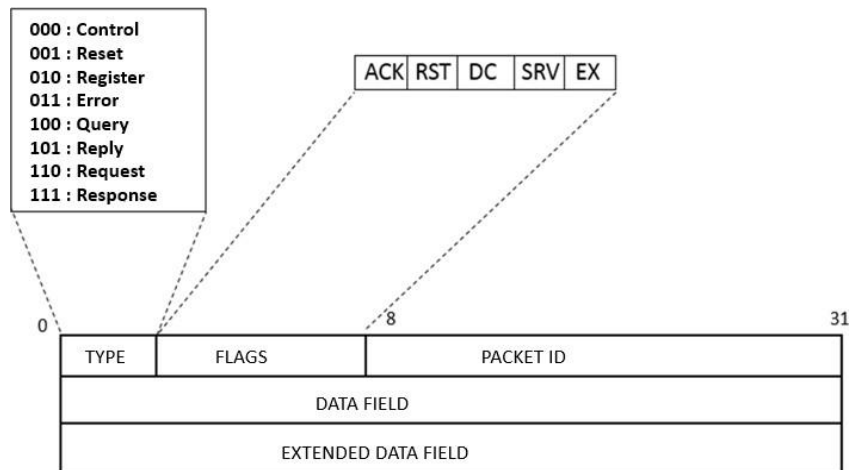


Figure 5. Packet header structure of Hybrid IoT Protocol

The 3-bit “Type” field in the header information shows the packet type. The 5-bit “Flags” field stores flags that are necessary for the proper functioning of communication. The 24-bit “Packet ID” field contains the unique ID of the packet.

The “Data Field” is a 32-byte long field that allows data from sensors to be transported in applications. “Extended Data Field” refers to an additional data transport area of 1024 bytes, which has been developed to support larger data transfer and operates by the “EX” flag.

The ACK flag is used for confirmation packets in communication. The RST flag is used to reset the direct access authorization through the IoT Gateway, which is used with the Reset packet. IoT Gateway, which receives the packet with the RST flag set, will lose direct communication. Thus, the coordinator server will be used as a tool to get information from the service. If the DC flag is “1”, direct communication to the relevant service can be provided. The SRV flag indicates server-generated traffic. The last flag, EX, indicates whether to use the Extended Data Field.

3.4 Working Phases of the Protocol

In this section, the phases of the developed protocol will be discussed in detail. Each step will be illustrated with packet structures and sample content.

3.4.1 Service Registration Phase

The purpose of this phase is to create a local database where services provided by IoT Gateways and access information to these services are kept. In cases where the number of sensors that need to be registered to the system is small, manual recording method or semi-automatic recording system can be used. However, in cases where this number is much higher, the manual registration method is inefficient, expensive and will require additional effort; in most cases it will be impossible to implement [27]. For this reason, a mechanism for semi-automatic recording of the services provided by sensors connected to IoT Gateway has been developed. Services such as temperature, light intensity, humidity provided by the sensor are combined with location information and recorded in the local database of IoT Coordinator.

The automatic registration of the sensor connected to the IoT Gateway to the local Coordinator device takes place at this phase. At this phase, the service provided by the IoT Gateway device, the IP address, whether it supports direct communication (DC) information, and the caching time is sent to the IoT Coordinator.

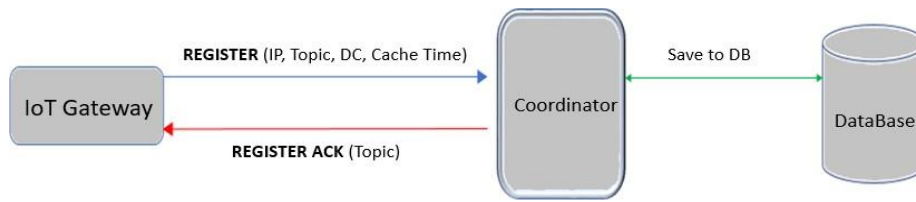


Figure 6. Service registration process of the developed protocol

For the “temperature” service provided in Lab1 location, the Register process is shown in Figure 6.

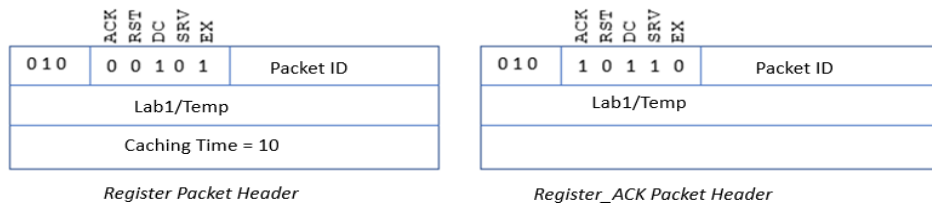


Figure 7. Packet headers of Register and Register_ACK

According to the example in Figure 7, the “Lab1/Temperature” service supports direct communication and the information sent by this service is kept in the cache of the coordinator server for 10 seconds. During this period, all data requests for the same service are retrieved directly from the coordinator cache without being interrogated to the sensor node again.

The server that receives the topic, cache time, DC information and the IP address of the IoT Gateway is saved to the local database. Following the registration, the registration confirmation packet (Register_ACK) is sent by the coordinator to the IoT Gateway.

During the service registration process of the developed protocol, all information is kept in the database of the Coordinator. Since this causes a single point of failure, service discovery queries will not be answered if the coordinator is failed. In order to prevent this problem, Automatic Registration feature has been developed for IoT Gateway devices. In the case of the server failure, the IoT Gateway sets itself to respond directly to service discovery requests if the register packet is not answered within a specified time period.

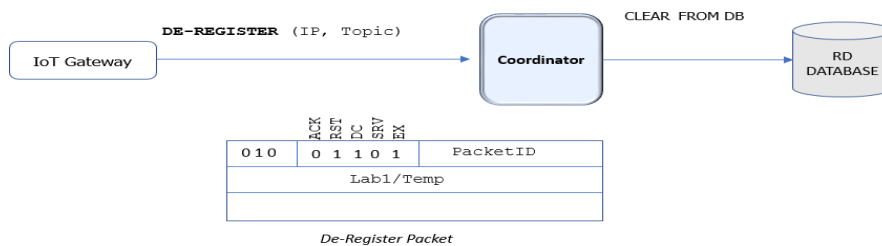


Figure 8. De-Register process and packet structure

If the service provided by IoT Gateway is disabled, registration must be canceled. In this case, the gateway sends a special Register packet with the RST bit set for the respective service. The Coordinator receiving the packet deletes the access information for this service from the database and responds to the IoT Gateway with the Register_Ack packet. The example in Figure 8 demonstrate the de-register packet for the “Lab1 / Temp” service.

3.4.2 Service Discovery Phase

To fully exploit the potential for successful implementation of IoT, there is a need for seamless and automatic discovery of available resources and dynamic access to the IoT device. MQTT does not have service discovery, but the CoAP protocol uses Uniform Resource Identifiers (URIs) for service discovery [28]. In the proposed protocol, the client queries the “topic” information for the service that it wants to access with “Query” packets. In the communication between the client and the server, the query regarding the location and service needed by the client is shown in Figure 9.

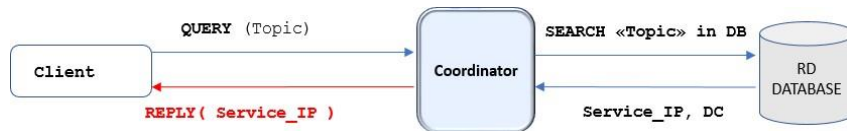


Figure 9. Service discovery process

The data related to the topic queried by the client is searched in the coordinator database, then the IP address of service notified to the client. If the relevant service supports direct communication, the IP address of the IoT Gateway is sent as a reply. Otherwise, the IP address of the Coordinator is sent.

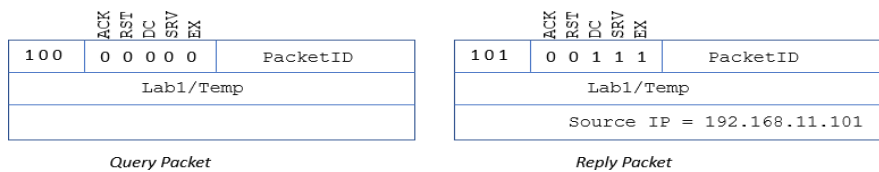


Figure 10. Query and reply packet headers

The example in Figure 10 shows the packet header of the “Query” sent by the client and the “Reply” packets given by the Coordinator in response.

3.4.3 Data Transfer Phase

At this phase, data request is made according to the “topic” information. Due to the hybrid nature of the proposed protocol, the data request can be provided directly from the Gateway, the sensor node (Figure 11), or via the Coordinator server (Figure 12).

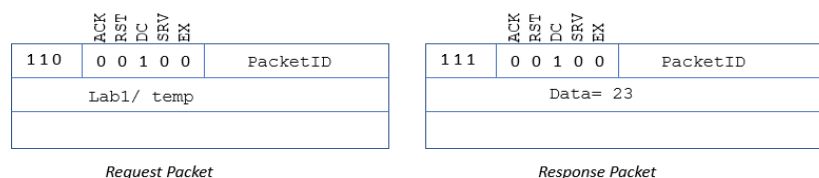


Figure 11. Process of direct access to the sensor node and packet headers.

During the Service Registration phase determines which methods of communication, either direct or through server, is supported. The packet structure and sample content for direct IoTGW access is given in Figure 11.

The Request message sent by the Client is given feedback via the Response packet that contains the data for the service being queried.

Modeling of server-based communication, another method supported by the hybrid protocol, is shown in Figure 12.



Figure 12. Data transfer process through server

In server-based communication, the client is not allowed to access the IoT Gateway directly.

Therefore, the data request from the service is made through the coordinator. The coordinator receives the Request packet from the client according to the “topic” and sends it to the IoT Gateway. The Response packet from IoT Gateway is sent to the client via the coordinator. In server-based communication, the sensor node only considers packets with the flag “SRV” marked “0” in the packet header.

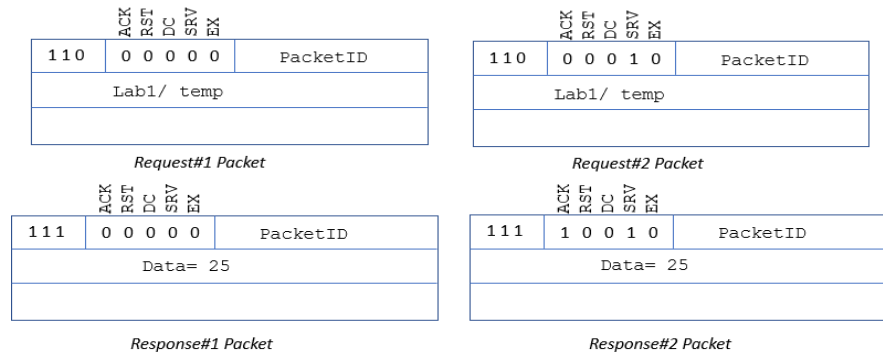


Figure 13. Request and response packets used in server-based communication

The header information and sample content of the packets used in server-based communications are given in Figure 13. When communicating through the coordinator server, the client has to connect to the server. All communication related to this service takes place through the coordinator. How long the data will be valid according to the cache time is determined during the service registration process. The coordinator stores data for the duration of the cache. During this period, other Request messages for the same service are answered directly from the cache without being sent to the sensor node.

3.5 Utility Packets

In addition to the five packets that offer the main functionality, Error, Reset and Control packet types are used to increase functionality.

Error Packet: It is sent to the client in case of any error in the local source database on the server, for instance, if there is no record of the subject queried, no access to the service, and no recordings are made to the database. The “error codes” for each error are also sent in the packet. An example of the error packet and the packet header structure is given in Figure 14.

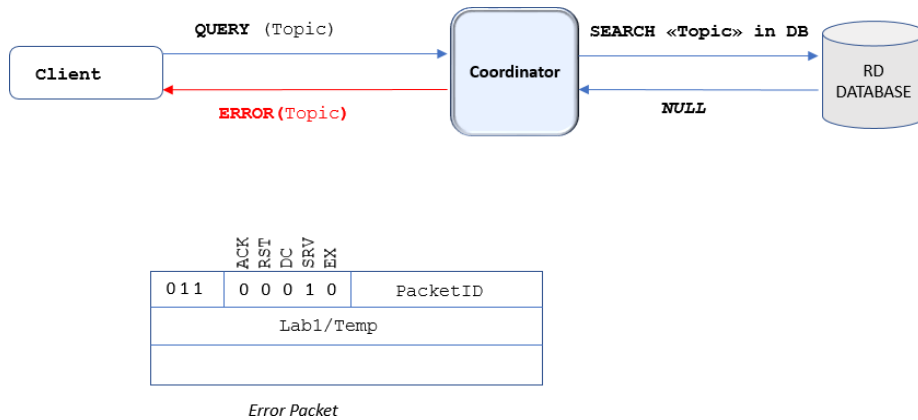


Figure 14. Error process and packet structure

Reset Packet : The type of access that the IoT Gateway device supports is determined during the Service Registration phase and this value remains constant. However, the direct access method may need to be revoked, depending on the state of the network. In this case, the Reset packet is sent by the coordinator to switch the communication method.

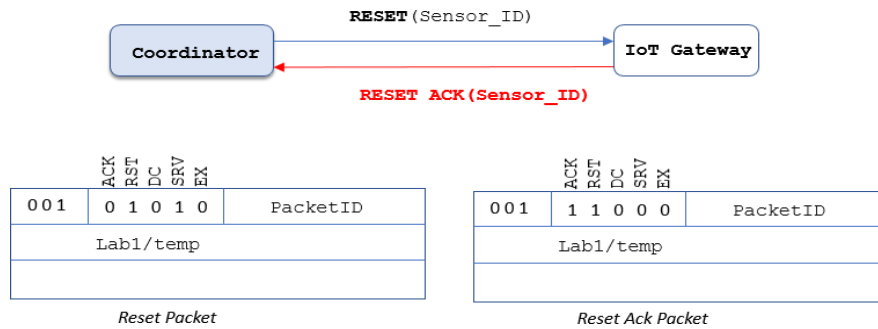


Figure 15. Process of changing (zeroing) the access method of the sensor node and packet structure

The communication method and packet structure of the reset packet is shown in Figure 15. The IoT Gateway that receives this packet loses its direct communication method.

Control Packet: Verification messages are used to check whether the service on the IoT Gateway is accessible. The control packets shown in Figure 16 are replied with control confirmation messages (Control_Ack).

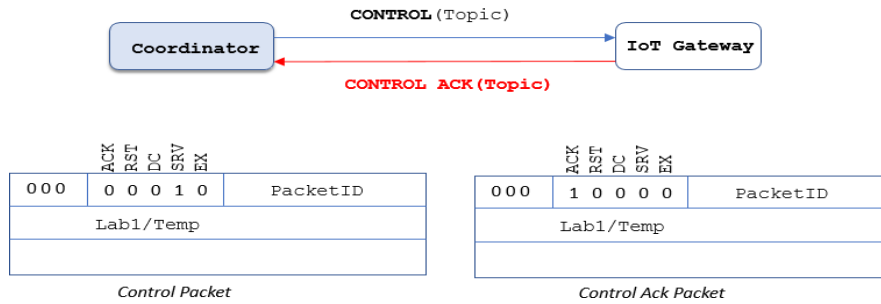


Figure 16. Control packet and sample header structure

Another advantage of the control messages is that the access time to the service can also be calculated by means of the confirmation messages. If no response to the control messages is received, the relevant service is assumed to be disabled and deleted from the coordinator's database. Likewise, when the Control messages do not reach the IoT Gateways, the coordinator is assumed to be failed and the sensor nodes (IoT Gateway) goes direct communication state.

3.6 Components in the Developed Protocol

As mentioned in the previous section, the proposed architecture of the protocol includes three components: Coordinator, Client software and IoT Gateway node. In this section, the working principles of these components are expressed in flow diagrams.

3.6.1 Client

An application that requests data about a location and service on the network and wants to interact with the sensor. It resembles the client that subscribes to any topic in the MQTT and AMQP structure. In the MQTT, the side that initiates the traffic is the publisher, and generally the data of the sensor is sent to the subscriber, regardless of whether the subscriber needs the information at the time. However, in the recommended architecture, the client side initiates traffic. As soon as data is needed, the data request is initiated by the client.

The client makes a query about how to access the service specified by “topic”. The query response from the coordinator contains the access information of the service source. The client matches the "topic" and the access information and stores it in the cache. Then, when the data request is made for this service, it goes directly to the data request stage without the need to query again and waits for the response from the sensor node. Figure 17 shows the flowchart of the client software.

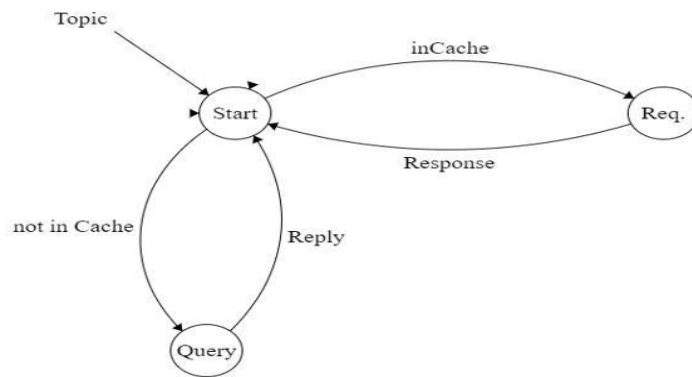


Figure 17. Client software flow diagram

The client software switches to query status or request status, depending on whether the “topic” information requested for data is in the cache and whether or not the Query response packet is received from the coordinator. The “topic” to be queried by the user is primarily searched in the cache. If there is a record of the subject information being queried in the cache, the IP address of the service is retrieved from the cache. Thus, the request status, which is the data transfer phase, is started. However, if there is no record for the requested topic in the cache, the service provider IoT Gateway must query the IP address and enter the query state.

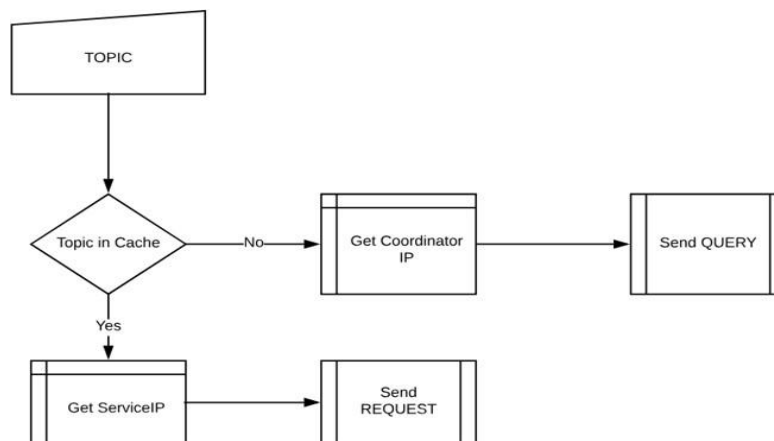


Figure 18. Client software initial state flow diagram

The flowchart for the initial state of the client software is given in Figure 18. The Send Query Packet and Send Request Packet processes in the diagram are presented as separate flow diagrams for modularity.

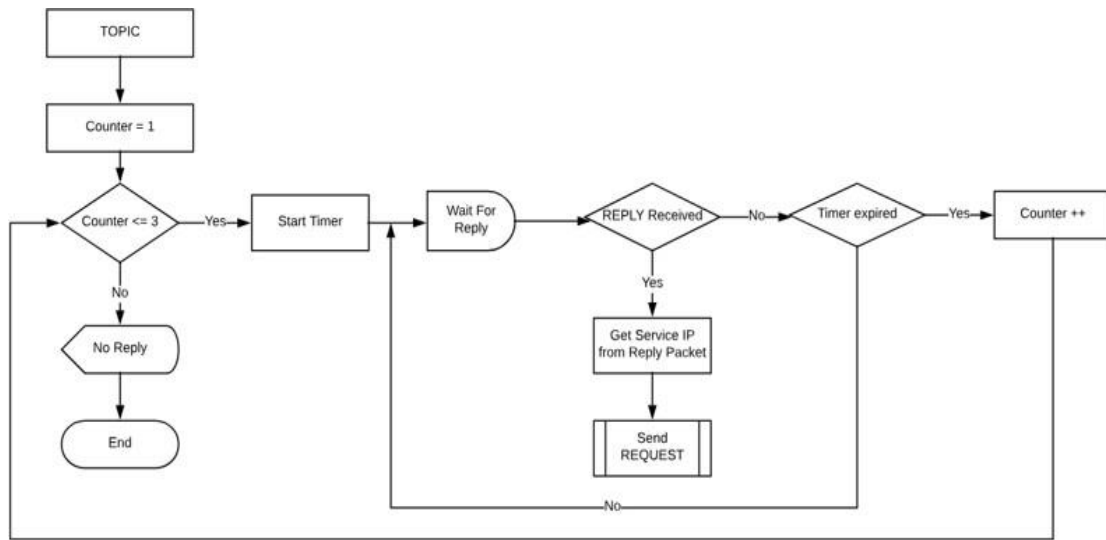


Figure 19. Flow diagram of query sending

The query process according to the “topic” information is performed by the Query packet sent from the client. The flow diagram of this process is shown in Figure 19. The query response message given to the query contains the service IP address. This information is stored in the client's cache for later reuse. As can be seen from the flow diagram, a certain time is expected for the query. If the Query Answer is not received within this period, the process is repeated 2 times. If the query is not received, the software is informed that the reply cannot be received. When the response is received, the client software switches to Request Send status.

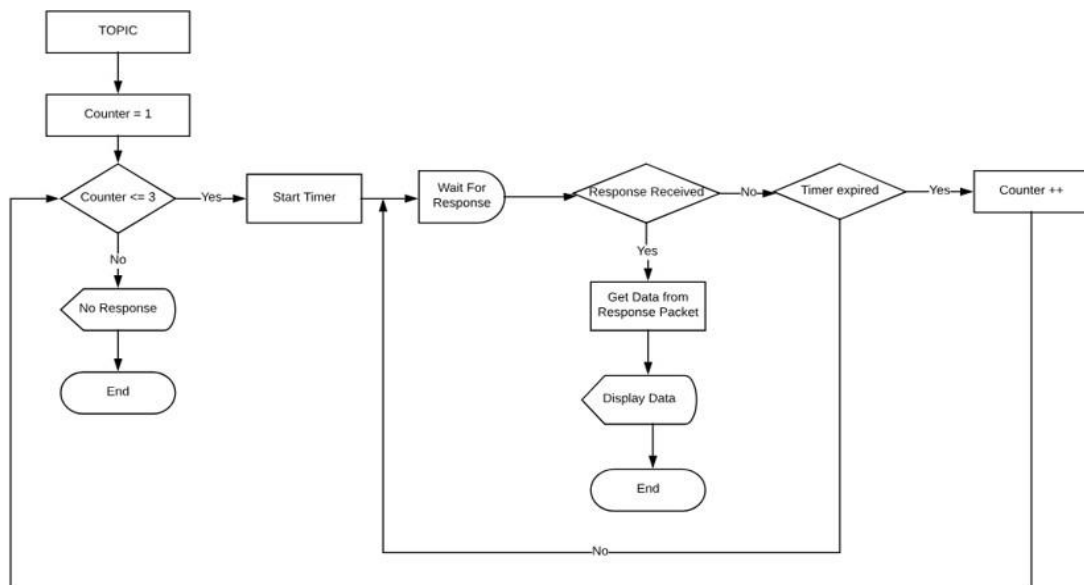


Figure 20. Flow diagram of request submission

On request, information from the IoT Gateway or Coordinator is included in the Response packets. As in the case of query, the request is also repeated 3 times. There is a 2 second dwell time for each operation. When the Response packet is received, data is extracted from the packet and sent to the client software. Otherwise, “no response” message is given to the software. This process is illustrated by the flow diagram in Figure 20.

3.6.2 IoT Coordinator

The most critical task in the proposed protocol is in the coordinator component. This component is responsible for listing the services available in the IoT ecosystem and for mediating users who want to access them.

One of the major problems encountered in the service discovery query on the IoT is the need to define a resource directory that is queried at the local level [22]. In the designed protocol, the coordinator keeps the service resources and how to reach these services in order to meet this need. It directs client traffic to the service source according to service discovery queries from clients. This device not only serves as a resource for service discovery, but also acts as an intermediary for certain types of services defined in the registration process. Accordingly, the flow diagram describing the functions of the coordinator device is shown in Figure 21.

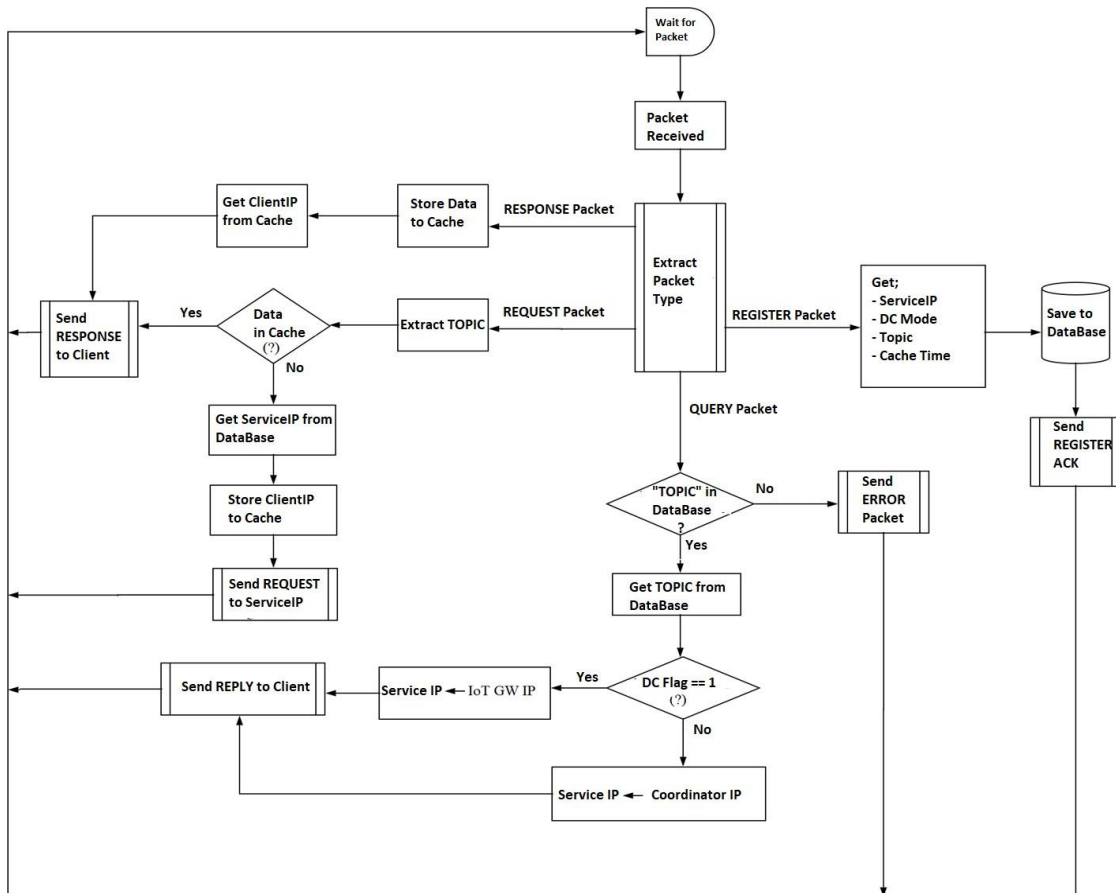


Figure 23. Flow diagram of IoT Coordinator

Identifying the incoming messages to the coordinator and the tasks to be performed according to the packet type are provided by the packet type analysis module. The tasks to be performed for each packet type are shown in the flow diagram in Figure 21.

3.6.3 IoT Gateway

Sensors, which are frequently used in the IoT, are devices with limited or no computational capabilities [21]. Therefore, in order to process the data obtained from the sensors, there is a need

for devices called IoT Gateway which can communicate with these devices, send and receive signals, and present the received signals to the network environment. The IoT Gateway concept is one of the critical components in the IoT [29]. This component acts as a proxy between the sensor network and the application layer. Sensors, the digital interfaces of objects in the IoT ecosystem, need these components to communicate with the IP network. In summary, IoT Gateways acts like a sensor node concentrator.

In the developed protocol, IoT Gateway (IoTGW) is responsible for encapsulating the data obtained from its sensors into IP packets and sending them to the client software. IoT Gateway supports both direct communication (DC) and access through the coordinator. Access method supported for each service is provided with the Register packet. . After registration, the client is ready to respond to requests (Request Packet). Figure 22 shows the finite state flow diagram of IoT Gateway.

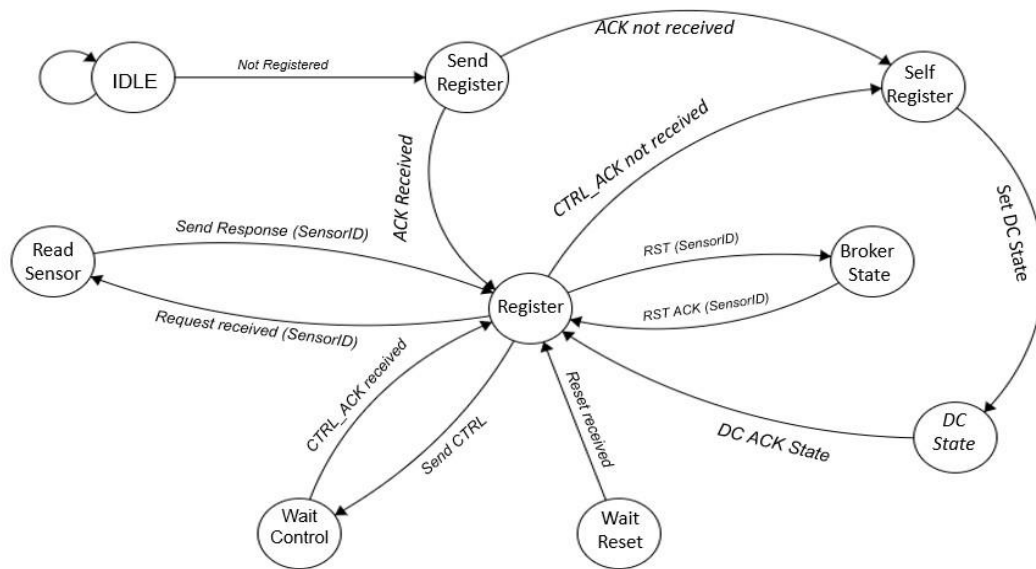


Figure 22. IoT Gateway flow diagram

As indicated in the diagram, it is confirmed by Control messages that the IoT Gateway is in constant communication with the Coordinator. However, the Coordinator creates a single point of failure in server-based communication. Therefore, it must be continuously verified whether the server remains disabled. Periodically sent Control messages check whether the server is accessible. If the server's accessibility is not provided, the IoT Gateway switches to Auto enrollment and supports direct communication ($DC = 1$). Thus, it can also respond to Query and Request packets that are queried by the client. According to the logic of the hybrid protocol, when the serving IoT Gateway is in direct communication state (DC), it loses direct access authorization from the coordinator and switches to server based operation mode (Server state).

3.7 Experimental Evaluation

For the performance evaluation of the developed protocol, a topology isolated from other network traffic was prepared in the laboratory and compared with the MQTT protocol which is frequently used in IoT applications. The simplified scheme of topology is shown in Figure 23.

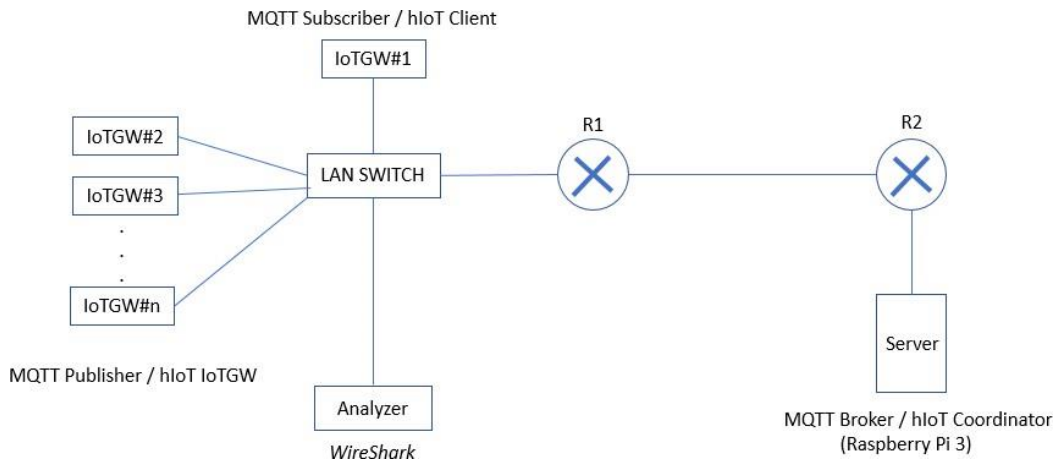


Figure 23. Experimental topology developed for performance testing

In the experimental topology, Arduino Unos were used as IoT Gateway devices and Raspberry Pi 3 as the server. The server was used as the "coordinator" in the hIoT protocol, and as the "MQTT Broker" in the MQTT architecture. Open source "Mosquitto" is used in MQTT broker. In the hIoT protocol, Python scripts are used on the server and SQLite is used as the database. In the topology, routers are used between the server and the IoT Gateway to measure the effect of different bandwidths. In the topology, a copy of all traffic sent and received by IoTGW was captured with WireShark to obtain average values. In the topology, 6 Arduino, 1 Raspberry Pi and 1 PC were used for analysis. Each communication was repeated 30 times and the mean values were calculated.

According to the information obtained from the captured packets, the average latency for six different bandwidths were calculated. As can be seen from Figure 24, at low bandwidths, the hIoT protocol has a lower latency. However, when the bandwidth increases, the gap between the latency of both protocols is closed.

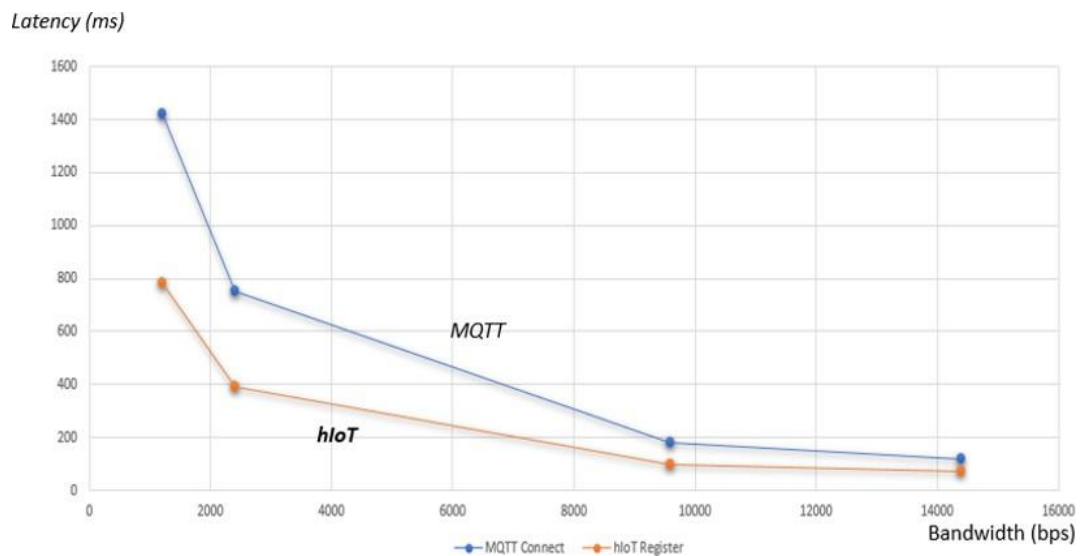


Figure 24. Latency in different bandwidths

In another analysis, latency based on payload were compared, during data transfer. In the end- to-end communication hIoT protocol has higher performance. However, MQTT showed higher performance in server based communication. The performance results are shown in Figure 25.

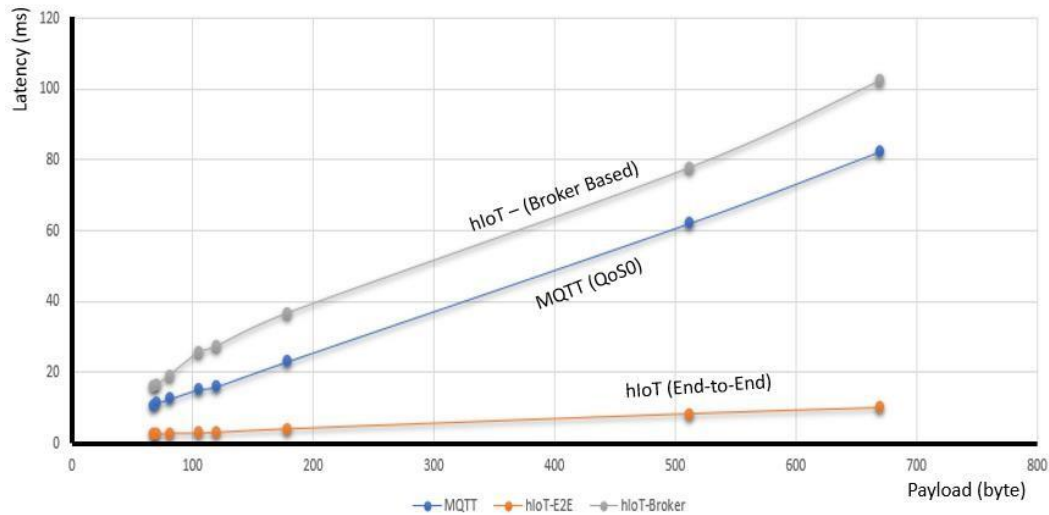


Figure 25. Latency in different payloads.

With the use of UDP in the proposed protocol and the simplicity of the packet header, it may be more suitable for devices with limited resources. Additionally, in applications where speed is important in the IoT ecosystem, the hIoT protocol can be used.

3.7.1 Limitations of Experiment

In this experimental testbed, evaluations were made on wired media. In order to evaluate the performance of the proposed protocol, tests should also be performed in wireless environments. In this study, the proposed protocol was compared only with MQTT protocol. Performance comparisons can be made with protocols used in the IoT ecosystem, such as CoAP. It will also be useful to make evaluations for multiple nodes using various simulation tools.

4. CONCLUSIONS

Data from objects in the IoT ecosystem differ according to applications. However, the data transferred in IoT applications are generally low and limited in size. The hIoT protocol, which was developed for the transmission of small data from sensors, is UDP-based and has a small packet header. Considering that the devices used in the Internet have low resources such as limited memory, processor and power, low overhead also contributes to efficient use of resources. Low latency times can be seen as an important criterion, especially in real-time applications. In the study of Jürgen et al., it is stated that UDP-based communications have lower RTT values in IoT environments than TCP-based communications.

In commonly used IoT protocols, information obtained from objects is sent periodically to the server via sensors. This information is again broadcast to the subscribers via the server. This means continuous use of the sensor and a relatively shorter life of the sensor and an increase in power consumption. In the architecture where the developed protocol will be used, optional communication is aimed. Thus, it is aimed to reduce the power consumption relatively.

In the hybrid protocol proposed within the scope of this study, there is no continuous dependence on the server in data communication and the devices in the network can communicate directly with each other. In this way, a single point of failure is prevented and traffic load is distributed and bottleneck formation is prevented. Again, in order to support the scenarios in which the central server is used, it is designed to be server-based. In cases where access to the data should be precise and instant, the sensors can be instantaneously communicated end-to-end via the sensors.

In heterogeneous IoT systems, access to each service is not equally important. Some services are critical, requiring direct communication, while others may compensate for this delay. Likewise, direct access to some services may involve security and performance risks, so it may be more appropriate to use middleware structures. This situation varies according to needs, usage scenarios, and security policies. In this study, a hybrid protocol has been designed to support various usage scenarios mentioned above in IoT systems.

In this study, whether the service supports direct access was determined during the service registration phase and it was seen to remain constant. However, instant decision making algorithms can be developed according to the determined traffic criteria. Again, according to various parameters, traffic estimation can be made and studies can be made to provide dynamic transitions between server-based or end-to-end communication.

Since most of the devices used in the IoT ecosystem have limited capacities, security features that require high memory and processing power, such as encryption, during the design of the protocol are excluded from this study. Considering that these limitations will gradually decrease with the development of technology, encryption, data integrity, and authentication studies can be performed in order to ensure secure communication in the protocol.

REFERENCES

- [1] A. Anjum, M. U. Ilyas, D. Gorden, C. Gar, and Guo et al, *Internet of Things – From Research and Innovation to Market Deployment*, vol. 6, no. 1. River Publishers, 2014.
- [2] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A Survey,” *Comput. Networks Int. J. Comput. Telecommun. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, *M2M and IoT Technology Fundamentals*. 2014.
- [4] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, “Part III. IoT Use Cases,” *From Mach. to Internet Things*, pp. 233–235, 2014.
- [5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “*Internet of Things (IoT): A vision, architectural elements, and future directions*,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [6] ITU-T, “Focus Group on M2M service layer: APIs and protocols overview,” *International Telecommunication Union*, 2014. [Online]. Available: <https://www.itu.int/opb/publications.aspx?parent=T-FG&selection=6§or>. [Accessed: 31-Oct-2019].
- [7] A. Talaminos-Barroso, M. A. Estudillo-Valderrama, L. M. Roa, J. Reina-Tosina, and F. Ortega- Ruiz, “A Machine-to-Machine protocol benchmark for eHealth applications - Use case: Respiratory rehabilitation,” *Comput. Methods Programs Biomed.*, vol. 129, pp. 1–11, 2016.
- [8] P. Bellavista and A. Zanni, “Towards better scalability for IoT-cloud interactions via combined exploitation of MQTT and CoAP,” *2016 IEEE 2nd Int. Forum Res. Technol. Soc. Ind. Leveraging a Better Tomorrow, RTSI 2016*, 2016.
- [9] J. Dizdarevic, F. Carpio, and A. Jukan, “*Survey of Communication Protocols for Internet-of- Things and Related Challenges of Fog and Cloud Computing Integration*,” *CoRR*, vol. 1, no. 1, pp. 1–27, 2018.

- [10] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of Things application layer protocol analysis over error and delay prone links," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop, ASMS/SPSC 2014*, 2014, vol. 2014-Janua, pp. 398–404.
- [11] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Integration of agent-based and Cloud Computing for the smart objects-oriented IoT," *Proc. 2014 IEEE 18th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2014*, pp. 493–498, 2014.
- [12] H. Kim, "Securing the Internet of Things via Locally Centralized , Globally Distributed Authentication and Authorization," 2017.
- [13] P. Thota and Y. Kim, "Implementation and Comparison of M2M Protocols for Internet of Things," in *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science Engineering (ACIT-CSII-BCD)*, 2016, pp. 43–48.
- [14] P. H. Su, C. Shih, J. Y. Hsu, K. Lin, and Y. Wang, "Decentralized Fault Tolerance Mechanism for Intelligent IoT / M2M Middleware," in *2014 IEEE World Forum on Internet of Things (WF- IoT)*, 2014, pp. 45–50.
- [15] N. Pathania, "Traffic Prioritization in an MQTT Gateway," in *International Journal of Computer Applications*, 2017, vol. 164, no. 2, pp. 32–38.
- [16] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings*, 2014, no. November.
- [17] D. Lars, "Performance Evaluation of M2M Protocols Over Cellular Networks in a Lab Environment," in *Conference: 18th International Conference on Intelligence in Next Generation Networks (ICIN), 2015, At Paris, France, 2015*, pp. 70–75.
- [18] J. E. Luzuriaga, M. Perez, P. Boronat, J. C. Cano, C. Calafate, and P. Manzoni, "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks," in *2015 12th Annual IEEE Consumer Communications and Networking Conference, CCNC 2015*, 2015, pp. 931–936.
- [19] I.-J. Shin, B.-K. Song, and D.-S. Eom, "International Electronical Committee (IEC) 61850 Mapping with Constrained Application Protocol (CoAP) in Smart Grids Based European Telecommunications Standard Institute Machine-to-Machine (M2M) Environment," *Energies*, vol. 10, no. 3, p. 393, 2017.
- [20] M. Collina, M. Bartolucci, A. Vanelli-coralli, and G. E. Corazza, "Internet of Things Application Layer Protocol Analysis over Error and Delay prone Links," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, 2014.
- [21] R. Klauck and M. Kirsche, "Bonjour Contiki: A Case Study of a DNS-Based Discovery Service for the Internet of Things," in *Ad-hoc, Mobile, and Wireless Networks*, 2012, pp. 316–329.
- [22] A. J. Jara, P. Martinez-Julia, and A. Skarmeta, "Light-weight multicast DNS and DNS-SD (ImDNS-SD): IPv6-based resource and service discovery for the web of things," *Proc. - 6th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. IMIS 2012*, pp. 731–738, 2012.
- [23] D. Choi, J. Jung, H. Kang, and S. Koh, "Cluster-based CoAP for Message Queueing in Internet- of- Things Networks," pp. 584–588, 2017.
- [24] C. Huo, "A Centralized IoT Middleware System for Devices Working Across Application Domains Using Self-descriptive Capability Profile," University Of California, 2014.
- [25] Y. Chen and T. Kunz, "Performance evaluation of IoT protocols under a constrained wireless access network," in *2016 International Conference on Selected Topics in Mobile and Wireless Networking, MoWNeT 2016*, 2016.
- [26] A. H. Ngu, M. Gutierrez, V. Metsis, and Q. Z. Sheng, "IoT Middleware : A Survey on Issues and Enabling Technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, 2017.
- [27] B. Kang, D. Kim, and H. Choo, "Internet of Everything: A Large-Scale Autonomic IoT Gateway," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 3, no. 3, pp. 206–214, 2017.
- [28] S. M. Kim, H. S. Choi, and W. S. Rhee, "IoT home gateway for auto-configuration and management of MQTT devices," *2015 IEEE Conf. Wirel. Sensors, ICWiSE 2015*, pp. 12–17, 2016.
- [29] A. Grygoruk and J. Legierski, "IoT gateway – implementation proposal based on Arduino board," in *Proceedings of the Federated Conference on Computer Science*, 2016, vol. 8, pp. 1011–1014.

AUTHORS**Erdal ÖZDOĞAN**

Received B.Sc. degree from Ankara University Astronomy and Space Science, and second B.Sc. degree, from Anadolu University Management Information Systems, Turkey. M.Sc. degree from Gazi University, Computer Education, Turkey. He is a PhD candidate in Information Systems at Gazi University. He gives trainings to public institutions and organizations on Computer Networks, Cyber Security and IoT. He is still a mathematics teacher at Ministry of National Education.

**O.Ayhan ERDEM**

Received B.Sc., M.Sc. and PhD. degrees from Gazi University Institute of Science and Technology, Turkey. In 1990 he attended to English Language Education Program of Indiana University, USA. He finished Technology of Computing Education at Purdue University. He has books and papers about, also he is doing research in the fields of Computer Networks, Programming Languages, IT, Computer Systems. He still is a Professor at Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Turkey.



SEGMENTATION OF SINGLE AND OVERLAPPING LEAVES BY EXTRACTING APPROPRIATE CONTOURS

Rafflesia Khan and Rameswar Debnath

Computer Science and Engineering Discipline, Khulna University,
Khulna, Bangladesh.

ABSTRACT

Leaf detection and segmentation is a complex image segmentation problem as leaves are most often found in groups with natural background. Edges of leaves cannot be clearly defined from image because of their color similarities. Also, separating every single as well as overlapping leaf individually is even more challenging as leaves share almost same color, texture and shape. In this paper, we propose a new automatic approach for leaf segmentation from image. Our leaf segmentation process uses efficient techniques for processing an image to obtain contours of every individual objects. Then, it selects the best appropriate connected contours that represent region of every leaves appearing in an image. Our model archives an overall 90.46% segmentation rate where segmentation rates for single and overlapping leaves are 95.34% and 86.73%, respectively.

KEYWORDS

image processing, leaf object segmentation, overlapping leaves, connected contour, object boundary detection.

1. INTRODUCTION

Plants are one of the most essential parts of nature and human lives. As almost every plant is identified by its leaf, proper plant-leaf identification is essential for agricultural productivity as well as industries such as drug, chemical, cosmetics, etc. Leaf identification is also used in crop disease identification and identification of rear as well as endangered plants. With the help of image processing and object detection based automatic models now a days we do not need experienced botanists and hedge effort for leaf identification task. Before identifying a leaf from image, using an automatic model, finding its location and segmenting the leaf region from image are the initial tasks. Leaf segmentation includes two major procedure: (1) segmenting foreground leaf region from natural background and (2) segmenting each single leaf and each occluded or overlapping (i.e., object on object) leaf individually from image. Figure 1 shows these two procedure of leaf segmentation process on an example image. Detecting leaves from complex natural background and separating every occluded leaves of same color and texture make leaf segmentation problem challenging. So, now a days, leaf segmentation from image is an area of growing research interest with significant applications.

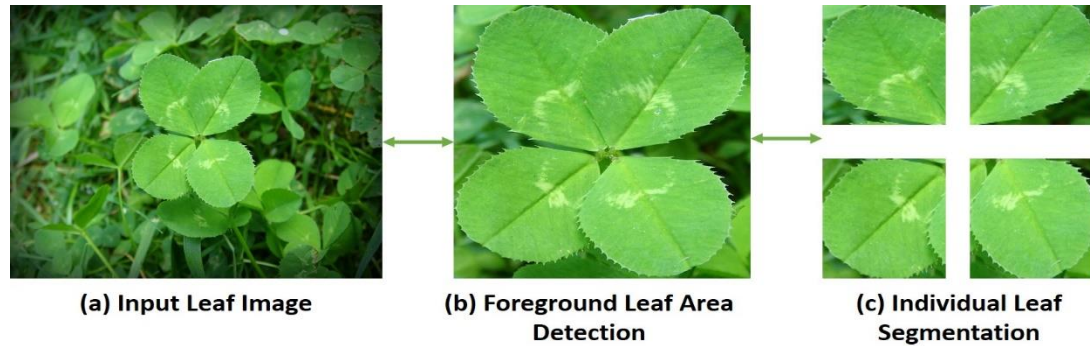


Figure 1. Major procedures of leaf segmentation.

Considering the significant importance and applications, numerous effective methods for leaf segmentation have been proposed [1] since the 1980s. The frequently used image object segmentation methods include edge-based, cluster-based, region-based and deep learning based methods. Niu et al. [2] proposed a model for cotton leaf segmentation using improved watershed algorithm. Dornbusch and Andrieu [3] developed a thresholding algorithm for estimating winter wheat's lamina boundaries. Combining global information with local statistical information Peng et al. [4] introduced a Chan Vese model for boundary detection of given leaf images. Although these models give good performance in case of segmenting a single leaf, accurate and non-destructive leaf segmentation is still a difficult task. These difficulties are caused by the uncertainties of overlapping condition and complex natural background of leaf surroundings. Considering the still existing complexities of leaf segmentation, number of segmentation techniques needed to be combined. Z. Wang et al. [5] presented an overlapping leaves image segmentation technique based on the Chan Vese model and Sobel operator. In [6], Cerutti et al. retrieved the leaf contour of image from a complex natural background by applying a two-step active contour algorithm using polygonal leaf model. Kenta Itakura and Fumiki Hosoi [7] proposed retrieval of plant structural parameters and methods for automatic and accurate leaf segmentation using 3D information point-cloud images. They combine distance transform and watershed algorithm. Chunlei, Wand et al. [8] used mean shift segmentation for segmenting foreground leaf region. Then they have implemented automatic initialization of active contour model (ACM) by calculating the center of divergence (CoD) and finally segmented occluded leaves individually using ACM. Daniel D. Morris [9] proposed a pyramid convolutional neural network with multi-scale predictions that finds and discriminates leaf boundaries from interior textures. Then using a watershed-based algorithm they estimate closed contour boundaries around individual leaves using previously detected boundaries. A comparative study of 14 unsupervised and 6 supervised segmentation models using Pl@ntLeaves dataset [10] was shown in [11].

Existing models combined with various segmentation techniques performs well in case of segmenting both single and overlapping leaf. But, there exists some still unsolved issues. Some models like [5], work with only one species of leaf which does not ensure the performance of model for leaves of different species found in different circumstance. Also, in real-life, randomly captured image can compromise the performance of some models [7, 8] that usually works with high resolution image captured with powerful camera. On the other hand, deep learning based models like, [9] do not work better while detecting leaves with internal texture and weak boundary clues. Total 20 models compared in [11], work for single leaf segmentation where all test images are captured focusing a target leaf at image center. But, in general leaves are most often found in groups where image foreground might contain multiple leaves overlapping one another.

Analyzing all these still existing complexities, in this paper, we propose a contour selection based leaf segmentation approach using various image processing techniques. In this model, our main

aim is to find the region of every individual leaves from image by detecting their appropriate contour. By contour, we indicate the outline that is marking the whole boundary of an object. Also, in this paper our considerable object is leaf. Most of the traditional contour detection algorithms of image either detect contour of leaf region from both foreground and background or detect contour without separating individuals. So, in our proposed model we apply some image processing techniques as pre-processing tasks before contour detection. These image processing techniques not only separate the foreground region from the background but also makes every individual leaf boundaries much easier to detect. So, from the processed image we can detect the contours that best represent all leaf regions individually. Finally, we segment those detected leaf regions (i.e., contours) as individual leaf images.

2. PROPOSED METHODOLOGY

Our proposed model works for segmenting every single as well as overlapping leaves separately. So, our goal is to find every contour (i.e., closed boundary) that precisely represents the outline of all visible regions of leaves in an image. Figure 2 shows the workflow diagram of our proposed model with an example image which visually shows the effect of every step on the input image. On the input image, at first we perform some preprocessing (i.e., section A in Figure 2). Leaf images can be of different types such as image with really complex background with multiple leaves or image with simple background with single leaf etc. Different types of image need different processing for better segmentation. So, we perform two different types of processing (i.e., section B and section C in Figure 2) on the result from A and generate two processed images. Next, we detect two sets of contours from both of those processed images and select one best contour set. Finally, we segment those best contours as individual leaf region from original input image (i.e., section D in Figure 2).

2.1. Preprocessing (Section A)

The preprocessing steps of our proposed model mainly works for preparing an image for leaf boundary detection. This is done by highlighting boundary edges and eliminating unnecessary internal and external edges of leaves.

2.1.1. Input Image

At first the input image is read in BGR (Blue, Green, Red) color format. As we use Python language and OpenCV library for the model implementation, the input image is read in BGR color format instead of RGB. The original version of the input image is stored as *Main image* and a copy of that image is used for further processing. Here, we store Main image because, after all processing finally our model segments or detects the actual contour of leaves appearing in input image from the original version of it. By this the system ensures noise free output images.

2.1.2. Resize

Processing big sized input image requires unnecessary power and computation time. It also sometimes hampers the segmentation accuracy. So, we select an optimal image size, 800×700 pixels, for input image. As the size ensures better performance for all our test experiments, after several number of test cases we have selected this size. When input image exceeds this optimal size, our model resizes it to the optimal size and then the resized image is stored as Main image. Otherwise, this step is ignored.

2.1.3. Preserve Edges

Every leaf has some internal textures and some species have even complex ones (e.g., African Blue). These textures create so many unnecessary edges which can manipulate target leaf area segmentation. In our model, for segmenting each leaf separately we wish to eliminate all unnecessary edges from image and preserve only the boundary edges.

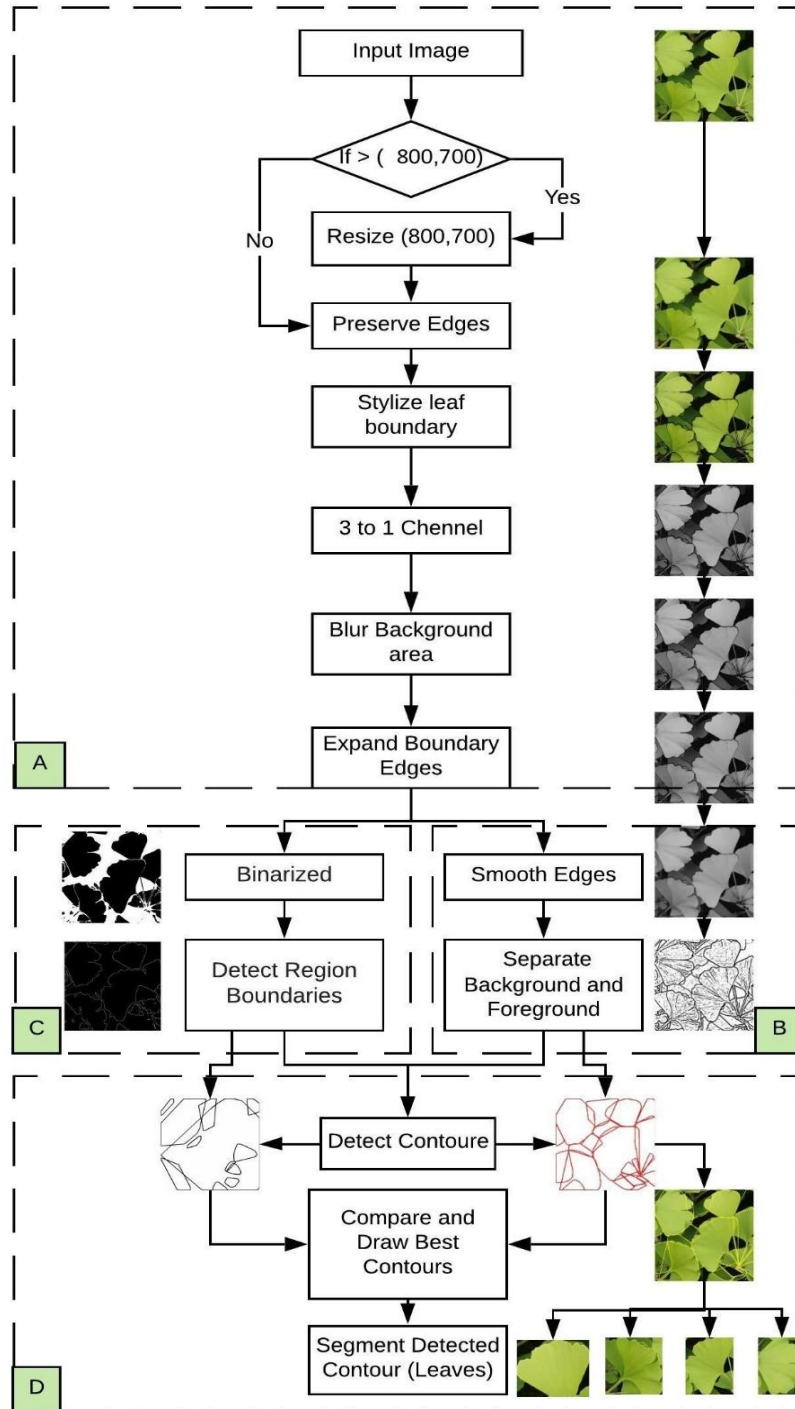


Figure 2. Workflow diagram of our proposed model with example image.

In this paper by unnecessary edges we refer all internal and external edges except the outline edges of every leaf object. Because, our aim is to find the contours or outlines of every leaf and

segment the connected regions within those. At this stage for smoothing internal texture and preserving boundary edges we apply Edge Preserving Filter [12] on image. As the required parameters we use 3rd edge preserving filter flag, value 40 for sigma_s (i.e., Sigma Spatial which controls the amount of neighborhood for smoothing), and value 0.3 for sigma_r (i.e., Sigma Range within the neighborhood which controls how dissimilar colors will be averaged).

2.1.4. Stylize Leaf Boundary

Our model draws thick boundary on the resulting image after smoothing internal texture and preserving boundary. We apply Stylization Filter to produce a watercolor effect on image which makes every object edge or outline or contour smooth and at the same time sharp [13]. It uses Normalized Convolution (NC) filter for providing better accuracy and works faster than other outline sharpening filters [14]. Eventually it thickens every outline edges and does not get affected by internal texture. For this procedure, the value of sigma spatial is set to 60 and the value of sigma range is set to 0.07.

2.1.5. Multi-Channel to Single Channel

The next step is to convert the BGR format (3 channel) image into grayscale image (1 channel) using equation (1). In grayscale image we need to process only one-third of the image data compared to BGR image which significantly reduces the amount of computation and memory consumption.

$$G_{g(i,j)} = 0.114I_{B(i,j)} + 0.587I_{G(i,j)} + 0.299I_{R(i,j)} \quad (1)$$

In equation (1), G represents the gray image and B, G, R represent the Blue, Green and Red channel respectively of image I, and i, j represents the coordinate value in x and y direction respectively.

2.1.6. Blur Background Area

On the stylized grayscale image, the next step is to eliminate the background. But, leaves are most often found in groups and our model works for segmenting multiple leaves individually. Our model does not use any direct background elimination algorithms as most of those target the center of image as foreground which might eliminate some of the foreground leaves which are not within the center of image. In our model we apply Gaussian filter for blurring image background. The Gaussian kernel is defined in 2D using equation (2). This non-linear filter enhances the effect of the foreground pixels and gradually reduces the effects of pixels which are farther from the center [14]. Thus, we get an image with much blurry background area and smoother foreground area.

$$G_{au-2D}(i, j; \sigma) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{i^2+j^2}{2\sigma^2}\right)} \quad (2)$$

Here $G_{au-2D}(i, j)$ represents the Gaussian Kernel function where i, j represents the coordinate value in x and y direction respectively and σ satisfies the width of the Gaussian kernel. In our model we apply a 9×9 kernel with σ value 1.5 in both x and y direction.

2.1.7. Expand Boundary Edges

In our work, we have performed morphological dilation once using a 3×3 integer valued kernel. Basically, image dilation enlarges the boundaries of regions of foreground pixels which also fills the holes within those regions [15] and joins broken parts. In our model, this procedure grows or

expands the areas of bright regions which eventually enlarges the outline edges of leaves. This step of processing also contributes in separating overlapping edges. Also, a bigger or thicker edge separates two connected objects better than a thin one. The formula used in this dilation process is shown in equation (3).

$$D_a(i, j) = I(i, j) \oplus \text{kernel } (3 \times 3) \quad (3)$$

Here $D_a(i, j)$ represents the output image and $I(i, j)$ represents source image where i and j represents the coordinate value in x and y direction respectively.

2.2. Processing of Type One Image (Section B)

In real-life, randomly captured image do not have object at the center of image with blurred background. These kinds of image contains scattered foreground objects with complex background and more unnecessary edges. These edges make images complex to process and hamper contour detection. So, for these kinds of images we need to do some extra processing before detecting the contour of leaves and running segmentation.

2.2.1. Smooth Edges

In case of type one image, when dilation process enlarges the brighter edges it might enlarge some still existing unnecessary internal texture edges of leaf. To handle that situation after dilation we perform a smoothing operation. For this, we apply a 2D Convolution [16] filter where we use a 5×5 averaging filter kernel to convolve through the image and rapidly smoothen all existing intensity variations within image pixels.

2.2.2. Separate Background and Foreground

In our proposed model, we wish for preparing an input image perfectly ready for individual as well as accurate leaf contour detection. Through all the previous steps we eliminate internal texture edges, thicken outline edges and also smoothen image. At this step, we separate the foreground region's pixels with a single intensity from the pixels on background. For this reason instead of using just a threshold value we use Adaptive Thresholding [17]. It calculates a different threshold for different regions of the same image. Here we use Adaptive Thresh Gaussian as adaptive method which uses threshold value as the weighted sum of neighbourhood values where weights are Gaussian window [18]. Value 1 is set as the subtracting constant from the weighted mean in case of weighted sum calculation. To calculate a threshold value, 11 is used as the size of a pixel's neighbourhood. We also use inverse threshold as thresholding style and this thresholding helps our system to separate background and foreground and make the foreground objects much visible [13]. At the end of this step, we get a background eliminated binary threshold image which is ready for contour detection.

2.3. Processing of Type Two Image (Section C)

An image focusing objects at center with less complex background and less amount of leaves, is another type of image. Removing background and making it ready for contour detection is less complex than type one image. Within our experiments, we find that running these two kinds of images through a same procedure does not result in better segmentation performance. So, we perform both these two types of processing for every image, detect set of contours from each processed image and then select the best set of contour.

2.3.1. Binarize

In type two processing, at first we binarize the resultant dilated image found after pre-processing for foreground and background separation. For this binarization, we apply binary thresholding with Otsu's thresholding. As Otsu's thresholding automatically determines the threshold value that best describes the image, we apply it directly for background removing. But, it do not perform better in case of non-bimodal image. To handle that we use binary thresholding and Otsu's thresholding together where Otsu's method automatically determines the threshold value, pixels below the threshold is turned off and pixels above the threshold value is turned on.

2.3.2. Detect Region Boundary

For detecting the leaves as connected contours from image after background separation we look for region boundary or border extraction. Our model performs border extraction by subtracting dilation result from erosion result. Dilation and erosion mostly affect the pixels that exists between the foreground and background as well as the pixels that exists close to the boundary. The difference between the dilation and erosion generally yields all object's boundary which significantly helps the segmentation task and prepares the image for a perfect individual contour detection. This boundary detection is performed using equation (4), (5) and (6) sequentially.

$$D_a(i, j) = I(i, j) \oplus kernel \quad (4)$$

$$E_r(i, j) = D_a(i, j) \ominus kernel \quad (5)$$

$$B_o(i, j) = D_a(i, j) \setminus E_r(i, j) \quad (6)$$

Here $D_a(i, j)$, $E_r(i, j)$ and $B_o(i, j)$ represents the image after dilation, erosion and border detection respectively, and i, j represents the coordinate value in x and y direction respectively.

2.4. Contour Detection and Segmentation (Section D)

At this step we perform contour detection. We have mentioned earlier that two types of images need different processing for better contour detection. Hence, this contour detection is performed on both the output images found after type one processing and type two processing. From the input image shown in Figure 2, we can see that contour detected from image with separated background foreground (i.e., the processed image of type two processing) is much better than image with region boundary (i.e., the processed image of type one processing).

2.4.1. Detect Contour

Both the processing of image type one and image type two results a background removed, unnecessary edge removed and boundary extracted image. This is the target image of our model for which we perform all the previously mentioned processing. From this image, we detect the connected regions by detecting their contours. We perform this contour detection using OpenCV Library's findContours() [20] method. The method detects contours from a binary image using the Topological Structural Analysis [21] algorithm. Method findContours() finds connected regions and stores them as individual contours by following object's border or outline of an image. Broken or closely connected outlines inhibits findContours() from detecting multiple object region separately. Also, if overlapping object's edges are not previously separated as individual object edges findContours() will detect one single contour containing all overlapped objects within it. Because of these regions our model passes an input image through all those above mentioned procedures. Which eventually makes the input image a noise free, smooth image with background removed and sharp outline of every individual objects.

2.4.2. Compare and Draw Best Contour

The previous step gives us two sets of contours detected from processed image of type one and processed image of type two. So, it is time to select the best set of contours from these two. Figure 3 shows the comparison process for an example image. For this we simply calculate the area of each contour and store these two sets of contour areas into two individual array, say A and B, in descending order (i.e., from one set of contours the contour with largest area is stored at the first index of corresponding array). Next, we filter these two sets by removing contour areas less than a threshold value. Analysing all our collected test images, we found that the contour area of a leaf object is more likely to be greater than area value 150. So, we found this threshold value 150 optimal for all our experiments. Without filtering, sometimes the model might consider a non-leaf region (e.g., region within two leaves, example shown at the right most contour of set B in Figure 3) as leaf region. After filtering, we compare these two arrays. For comparison at first we check whether the number of contour areas of both A and B is greater than five or not. If both sets has more than five contours, we calculate the total amount of area of first five contours for both set A and B and compare the amount. The set with largest amount is selected as the best contour set.

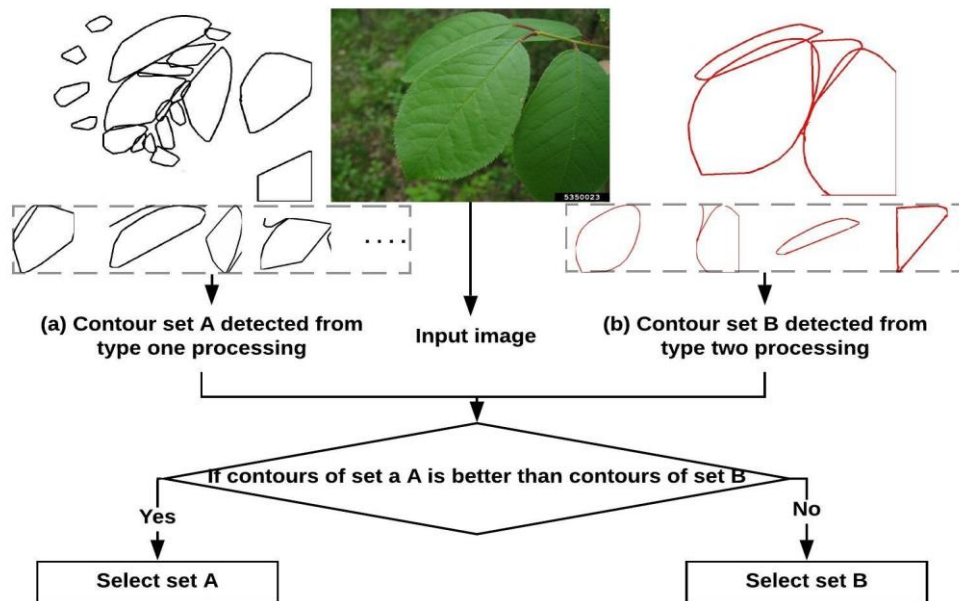


Figure 3. Comparison process between two sets of contours.

If any of set A or B has number of contour areas less than five, we check for the set which has less contour and store the quantity number, say x , of contour areas of that set. Then comparison and best set selection procedure are done as previously using x number of contour areas instead of five. After a large number of test cases we select five as the threshold value for this comparison and it performs well with our experimental dataset. But, we believe this comparison process can be improved with new processes and we are currently working on it. After contour selection, we draw the selected contours on a mask image which has the same size as Main.

2.4.3. Segment Detected Contour (Leaves)

Our model processes an input image following the above mentioned procedures and selects the best set of contours from the image. These contours actually represent the regions of leaves in

main input image. So, the final step is to segment these found contours as regions of image which are actually the regions of individual leaves in Main image.

From our selected set of contour on mask image, for each contour, we consider left-to-right as x direction and top-to-bottom as y direction. Thus we find a point say (a, b) as top left corner of that particular contour. Following that we get a point (say (a, b)) as the top-left vertex and another point (say (c, d)) as the bottom-right vertex for each and every contour region. Using these points we make a rectangular area surrounding each contour within it. Then using these vertex coordinates we crop corresponding region of mask image from the Main image. This cropping is just like array slicing. So, for cropping each contour area from image, we supply the b and d coordinates, followed by a and c coordinates to slice a rectangular portion from Main image that exists within the (a, b) and (c, d) coordinates' surrounding area. Then these rectangular portions are stored as individual images which represent the final leaves segmented from the Main input image.

3. PERFORMANCE AND COMPARATIVE ANALYSIS

In this section we discuss about the performance of our proposed leaf segmentation model along with some comparative analysis. There exists so many datasets for dense object detection or segmentation. But, most of the well-known leaf datasets contain only single leaf image having white or black background. To analyse the performance of our proposed model we need images with single as well as overlapping leaves in complex natural background. Hence, the experiments of our proposed leaf segmentation model are carried out on leaf images collected from the Internet and Pl@ntLeaves dataset [10]. Dataset [10] includes 233 images but most of them are center focused single leaf images in natural background. To ensure our model's performance we build a dataset containing 190 images of leaves where 153 images are collected from [10] and 37 images are randomly collected from the Internet. These images contain 150 single leaves i.e., leaves without occlusion and 196 occluded leaves i.e., leaves involved in occlusion or overlapping one another.

The performance of our proposed model is presented in Table 1. For explaining this comparison we follow the way explained in [8]. From all our collected 190 images we calculate the percentage of correctly segmented single leaves as well as overlapping leaves. Our model's leaf segmentation performance of every individual leaf is evaluated by segmentation rate and failure rate. In Table 1 the portion of leaves indicates overall 346 individual leaves are found within our collected 190 images, where 150 leaves are found single and 196 are found with occlusion. Segmentation rate refers to the proportion of correctly segmented individual leaves from the total number of individual leaves. Failure rate refers to the proportion of incorrectly segmented individual leaves from the total number of individual leaves.

Table 1. Segmentation performance for occluded and single leaves individually.

	Portion of Leaves	Segmentation Rate	Failure Rate
Single Leaves	43.35% (150)	95.34% (143)	4.66% (7)
Occluded Leaves	56.65% (196)	86.73% (170)	13.27% (26)
Overall	100% (346)	90.46% (313)	9.53% (33)

Our model successfully segments 313 leaves from 346 leaves of 190 images which ensures 90.46% overall segmentation rate. Within the 346 leaves 196 leaves were found with occlusion and our model correctly segments 86.73% of them (i.e., 170 out of 190). Table 1 also shows that

the rate of single leaf segmentation of our model is 95.34% and it only fails to segment 7 single leaves out of 150.

Analysis the failure rate of our model we found that in most of the cases our model fails to detect leaves whenever the input image is of very low resolution. Also, image being too much blurry hinders the segmentation process.

Within the 26 leaves with occlusion that our model fails to segment, some were over segmented and some were under segmented. The rate of over segmentation (e.g., one leaf segmented into several parts) and under segmentation (e.g., two or multiple leaves segmented as one) is 42.31% (11 out of 26) and 57.69% (15 out of 26) respectively. Figure 4 shows some example of segmented leaf regions with Over-segmentation (OS) and Under-Segmentation (US).

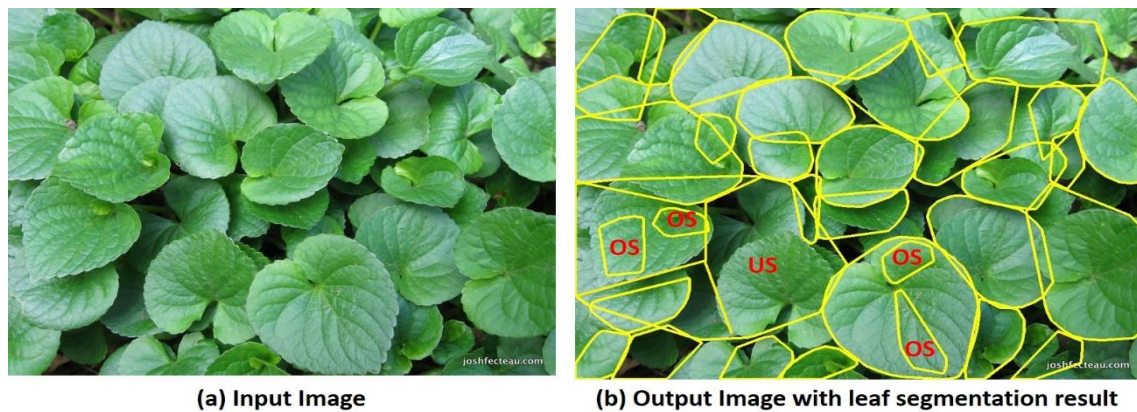


Figure 4. Model output with over segmentation and under segmentation.

The performance of our proposed model is quite satisfactory for some reasons such as: (1) our model can segment both single and occluded leaf individually at a high segmentation rate, (2) it can both separate the closely connected leaves (i.e., leaves touching each other closely) and the overlapping leaves (i.e., leaves overlapping one another), (3) it works better on different types of images as all 190 test images are collected from different resource, (4) it provides good accuracy in segmenting leaves of different species having different shape and texture.

Figure 5 shows how accurately our proposed model performs leaf segmentation procedure. Figure 5(a) shows an image with its final leaf segmentation result which is collected from our 37 Internet images and Figure 5(b) shows an image with its final leaf segmentation result which is collected from dataset [10] 153 images.

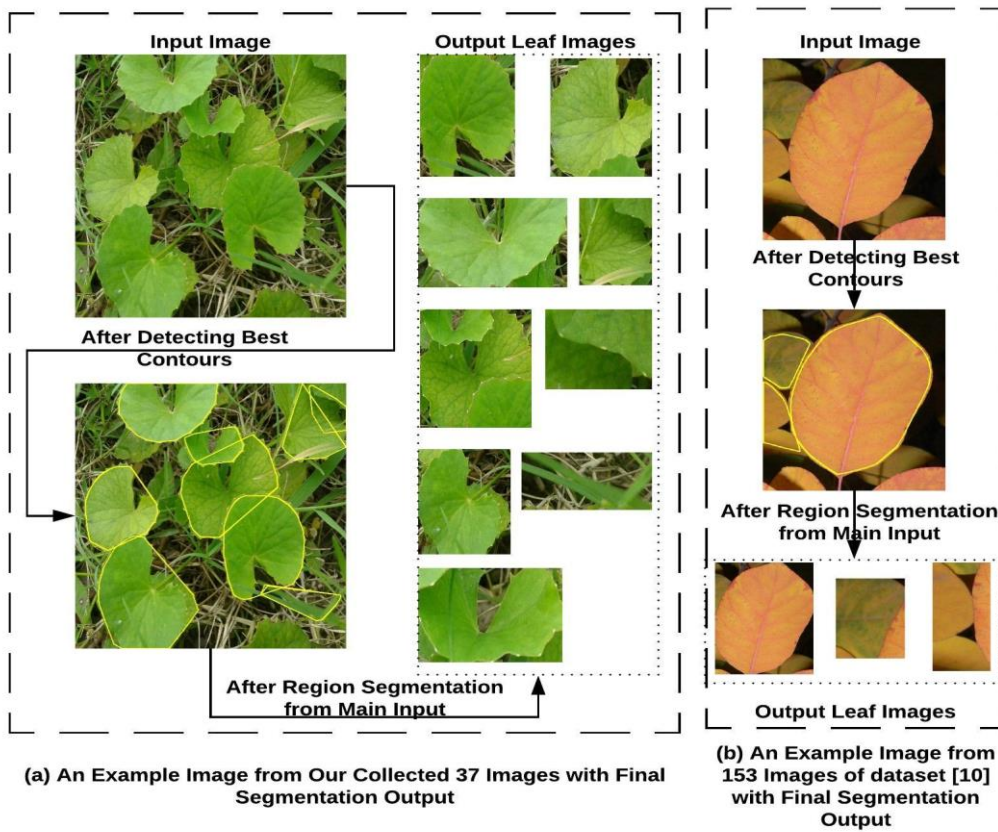


Figure 5. Leaf segmentation example of two images with input and output.

The combination of image processing and best contour selection approach of our model ensures better leaf boundary detection compared to some well-known edge detection algorithms that are usually used for object segmentation. Figure 6 shows a source image along with the output results found after applying watershed, canny, laplacian, sobel and our model’s processing on it. Figure 6 also shows the contours detected from watershed, canny, laplacian, sobel and our model’s processing applied resultant images which ensures that our model detects better contours, of the two overlapping leaves of Figure 6, individually compared to others.

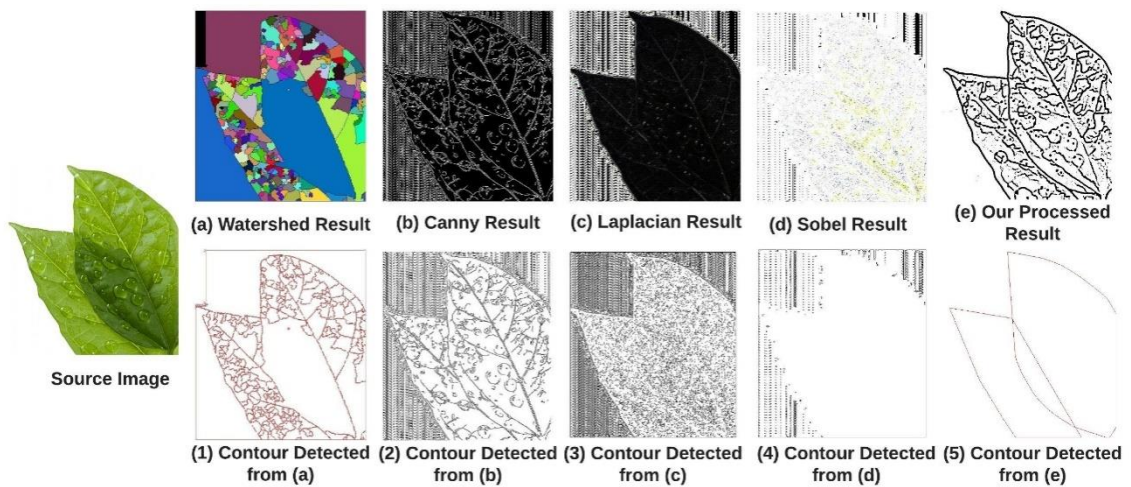


Figure 6. Comparing our model’s image processing and contour detection with some other algorithms.

Within our study, most of the existing leaf segmentation models work with different datasets and use different performance measures for evaluating their model. So, instead of comparing accuracy rate, we compare our model on some complex scenario where leaf segmentation becomes tough. Enlisting these scenarios we check whether our model can segment individual leaves, compared to other existing models, covering each scenario.

Table 2. Performance comparison of accurate segmentation on different scenario.

Some complex scenario of segmentation that existing models covers or not covers	Models					
	<i>Our model</i>	<i>Ref. [5]</i>	<i>Ref. [7]</i>	<i>Ref. [8]</i>	<i>Ref. [9]</i>	<i>Models of Ref. [11]</i>
Leaves without having much boundary clue	√	~	~	~	×	√
Image not focusing target leaf object	√	×	×	√	√	×
Image having overlapping leaves at any corner but not always at center of image	√	√	×	√	√	×
Every individual overlapping leaf from image	√	×	√	√	√	√
Image of leaves with different and internal texture	√	√	√	×	×	√

√=Leaf Segmentation Possible, ×= Leaf Segmentation Not Possible, ~ = Criteria not applicable for this model as authors do not clarify.

Table 2 shows a comparative analysis of leaf segmentation from image, of our model with some existing ones, on different complex scenario. It also explains that our proposed model ensures a proper and compelling leaf segmentation from image by covering various complex scenario.

4. CONCLUSIONS

This paper proposes a leaf segmentation system where we aim for segmenting single as well as occluded leaves individually from image. At first we apply some image processing techniques so that the outline or contour edges of every leaf becomes much visible, smooth and sharp which helps in individual object's contour detection. Then our model performs proper filtering and comparison to select the best contour set that matches with the shape of leaves. Finally, our model segments those selected contour areas as leaf regions from main image. From the experimental results, we see that our model archives an overall segmentation rate 90.46% while segmentation rates for single and overlapping leaves are 95.34% and 86.73% respectively, on our created dataset.

The working processes described in this paper is applied for detecting contour edges of overlapping leaves from complex background. In future we look forward to improve and apply these techniques on different sectors such as medical cell images where overlapping objects are found within low variance complex background. Our future works will also focus on improving the performance of this model even with less processing.

REFERENCES

- [1] Guyer DE, Miles GE, Schreiber MM, Mitchell OR, Vanderbilt VC. Machine vision and image processing for plant identification. Transactions of the American Society of Agricultural Engineers 1986;29(6):1500-1507.

- [2] Niu C, Li H, Niu YG, Zhou ZC, BU YL, Zheng WG. Segmentation of cotton leaves based on improved watershed algorithm. In: 9th International Conference on Computer and Computing Technologies in Agriculture, CCTA 2015, Beijing, China, 27-30 September 2015 .
- [3] Dornbusch T, Andrieu B. Lamina2Shape-an image processing tool for an explicit description of lamina shape tested on winter wheat (*Triticum aestivum* L.). *Computers and Electronics in Agriculture* 2010;70(1):217-224.
- [4] Wu P, Li WL, and Song WL. Segmentation of leaf images based on the active contours. *International Journal of u- and e- Service, Science and Technology* 2015;8(6):63-64.
- [5] Z. Wang, K. Wang, F. Yang, S. Pan, Y. Han, Image Segmentation of Overlapping Leaves Based on Chan–Vese Model and Sobel Operator, *Information Processing in Agriculture* (2017), doi: [https://doi.org/ 10.1016/j.inpa.2017.09.005](https://doi.org/10.1016/j.inpa.2017.09.005).
- [6] Cerutti G, Tougne L, Mille J, Vacavant A, Coquin D. Understanding leaves in natural images - a model-based approach for tree species identification. *Computer Vision and Image Understanding*, 2013;117(10):1482-1501.
- [7] Itakura, Kenta, and Fumiki Hosoi. "Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3D plant images." *Sensors* 18.10 (2018): 3576.
- [8] Xia, Chunlei, et al. "In situ 3D segmentation of individual plant leaves using a RGB-D camera for agricultural automation." *Sensors* 15.8 (2015): 20463-20479.
- [9] Morris, Daniel. "A pyramid CNN for dense-leaves segmentation." 2018 15th Conference on Computer and Robot Vision (CRV). IEEE, 2018.
- [10] H. Goëau et al., "The CLEF 2011 plant images classification task," Image CLEF 2011 working notes.
- [11] Grand-Brochier, Manuel, et al. "Tree leaves extraction in natural images: Comparative study of preprocessing tools and segmentation methods." *IEEE Transactions on Image Processing* 24.5 (2015): 1549-1560.
- [12] E. Gastal, "Non photorealistic rendering using opencv(python, c++) | learn opencv." [Online], Available: <https://www.learnopencv.com/non-photorealisticrendering- using-opencv-python-c/>.
- [13] Rafflesia Khan, Rameswar Debnath, " Multi Class Fruit Classification Using Efficient Object Detection and Recognition Techniques", *International Journal of Image, Graphics and Signal Processing(IJIGSP)*, Vol.11, No.8, pp. 1-18, 2019.DOI: 10.5815/ijigsp.2019.08.01
- [14] "Image Filtering Techniques in OpenCV." Packt Hub, 18 Apr. 2018, [Online], Available at: <https://hub.packtpub.com/image-filtering-techniques-opencv/>.
- [15] "Dilation." Morphology - Dilation, [Online], Available at : <https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>.
- [16] "Smoothing Images." OpenCV, [Online], Available at : https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html.
- [17] P. O. A. Thresholding, "Adaptive Thresholdings," (2003). [Online; last accessed 06-April-2019].
- [18] OpenCV - Adaptive Threshold, Tutorials Point, [Online], Available at: https://www.tutorialspoint.com/opencv/opencv_adaptive_threshold.htm

- [19] S. F. BogoToBogo_K Hong Ph.D. Golden Gate Ave, "Image thresholding and segmentation." [Online] Available: https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Global_Thresholding_Adaptive_Thresholding_Otsus_Binarization_Segmentations.php
- [20] Structural Analysis and Shape Descriptors - OpenCV 2.4.13.7 Documentation, [Online], Available at https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=drawcontours#drawcontours
- [21] Suzuki, Satoshi. "Topological structural analysis of digitized binary images by border following." Computer vision, graphics, and image processing 30.1 (1985): 32-46.

AUTHORS

Rafflesia Khan is a student of M.Sc. at Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh. She has completed her Bachelor's degree from Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh in 2017. Her research areas of interest are image processing, visual object detection & recognition, machine learning, pattern recognition, and internet of things security.



RameswarDebnath is a Professor of Computer Science and Engineering Discipline at Khulna University, Khulna, Bangladesh. He has completed his Bachelor degree from Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh in 1997. He has received his Masters in Engineering degree and Ph.D. degree in Computer Science and Engineering from the University of Electro-Communications, Tokyo, Japan in 2002 and 2005 respectively. His research areas of interest are image processing, statistical machine learning and its applications to pattern recognition, visual object detection and natural language processing.



SPLIT MULTI-STAGE VECTOR QUANTIZATION BASED STEGANOGRAPHY FOR SECURE WIDEBAND SPEECH CODER

Merouane BOUZID and Bakkar LASKAR

Speech Communication and Signal Processing Laboratory,
University of Sciences and Technology Houari Boumediene (USTHB),
Electronics Faculty, P.O. Box 32, El-Alia, Bab-Ezzouar, Algiers, 16111, Algeria

ABSTRACT

Speech steganography is a technique of covert communication which conveys secret speech hidden in cover digital speech signal in such a way that the existence of the secret speech is concealed. In this paper, we develop a steganographic speech coding system based on embedding coded secret speech into host public speech coded by the AMR-WB (ITU-T G.722.2) speech coder. For the compression of the secret speech signal, we used the 2.4 kbits/s MELP speech coder. The embedding process of the secret bit stream is carried out into the split-multistage vector quantization (S-MSVQ) indices of G.722.2 immittance spectral frequencies (ISF) by modifying the mechanism of the S-MSVQ second stage.

KEYWORDS

Multi-stage vector quantization, steganography, data hiding, ISF parameters, secure speech, wideband speech coder, MELP, AMR-WB

1. INTRODUCTION

Steganography is the art of sending secret information in a cover media without arousing suspicion. Indeed, modern steganography techniques exploit the characteristics of digital media by using them as carriers (covers) to hold hidden information. Thus, the sender embeds secret information in a digital cover file to produce a stego-file, in such a way that the contents of hidden data and its existence cannot be detected by an observer during the transmission process [1]. The secret information can be extracted only at the authorized user's side.

In this work, we focalize particularly on speech steganography techniques which consist in hiding a secret speech signal into a cover (host) signal. A variety of speech/audio steganography methods have been proposed in the past, where most of them are based on the temporal domain, the transform domain and the compression domain. An extended review of the current state-of-art literature in digital audio/speech steganography techniques in each domain is given in [2]. In compression domain, speech steganography techniques based on vector quantization (VQ) have been getting more and more popular, since they enhance the conventional VQ coding by adding the possibility of data hiding.

In [3], Chang and Yu proposed a dither-like data hiding method to embed hidden data in the multistage vector quantizer (MSVQ) of the Mixed-Excitation Linear Predictive (MELP) and ITU-T G.729 speech coders. In [4], Geiser and Vary developed a steganographic method to embed digital data in the bitstream of an ACELP speech coder. In [5], Laskar and Bouzid proposed two

variants of VQ-based speech steganography binning schemes (SBS) for G.722.2 secure speech communication system. They showed that the two steganographic SBS methods carried out by balanced and unbalanced VQ codebook partitioning can generate stego-speech signals with similar quality to cover speech signals. In [6], an AMR-WB speech steganography system was proposed based on diameter-neighbor codebook partition method. It was shown that speech steganographic system can provide higher and flexible embedding capacity without noticeable decrease in speech quality and better performance against statistical steganalysis.

Since the Adaptive Multi-rate Wideband AMR-WB (Rec. G.722.2) [7], [8] speech coder still a good candidate for cover medium in speech steganography, we develop in this paper a steganographic AMR-WB coding system based on the dither-like data hiding idea. It's about modifying the mechanism of the second stage of the split-multistage vector quantizer (S-MSVQ) of G.722.2 immittance spectral frequencies (ISF) parameters to embed a secret speech coded by the 2.4 kbits/s MELP [9] speech coder.

An outline of this paper is as follows. In section 2, we first review briefly the basics of the conventional VQ, the split vector quantizer (SVQ) and the MSVQ. Then, we present the dither-like data hiding method applied on the MSVQ scheme. In section 3, we describe the design principle of a steganographic G.722.2 speech coding system developed according to the S-MSVQ based data hiding method. Experimental results are provided in section 4 to evaluate the performance of our speech steganographic system. Conclusions are given in section 5.

2. DITHER-LIKE DATA HIDING ON MSVQ QUANTIZER

Several data hiding methods, based on conventional VQ, have been proposed in literature [1], [10]. One of the most popular quantization-based data hiding method is probably the quantization index modulation (QIM) [11].

Before presenting the dither-like data hiding method applied on MSVQ scheme, let us first review briefly the basics of the conventional VQ, the split vector quantizer (SVQ) and the MSVQ.

2.1. Basic principle of VQ, SVQ and MSVQ schemes

A k -dimensional VQ of rate R bits/sample (bps) is a mapping of k -dimensional Euclidean space \mathcal{R}^k into a finite codebook $Y = \{y_0, \dots, y_{L-1}\}$ composed of $L = 2^{kR}$ codevectors [12]. The design principle of a VQ consists of partitioning the k -dimensional space of source vectors x into L non overlapping cells $\{R_0, \dots, R_{L-1}\}$ (partition) and associating with each cell R_i a unique codevector y_i such that the total average distortion D is minimized [12]. Various algorithms for the optimal design of VQ have been developed in the past. The most popular one is certainly the LBG algorithm [12]. This algorithm is an iterative application of the two optimality (nearest neighbor and centroid) conditions such as the partition and the codebook are iteratively updated.

In other hand, an N part k -dimensional SVQ (noted N -SVQ) is composed of N classical VQs of smaller sizes and dimensions [13]. Its basic principle consists of partitioning the set of the training base vectors x of dimension k in N subsets of sub-vectors of smaller dimension k_i (with $\sum_{i=1}^N k_i = k$). Then, for each part, the corresponding VQ codebook will be designed by using the LBG-VQ algorithm. Compared to a conventional unstructured k -dimensional VQ, of rate R bps and size $L = 2^{Rk}$, an N -SVQ is thus composed of N codebooks of smaller sizes $L_i = 2^{R_i k_i}$ (where $L = \prod_{i=1}^N L_i$ and R_i is the partial rate in bps). Figure 1 shows a bloc-diagram of an N -SVQ quantizer.

Concerning the conventional MSVQ, we can say that it is a kind of cascaded VQ where the output of one stage is given as an input to the next stage and the bit rates used for quantization are divided among all successive MSVQ stages [14], [12]. The first MSVQ stage performs a

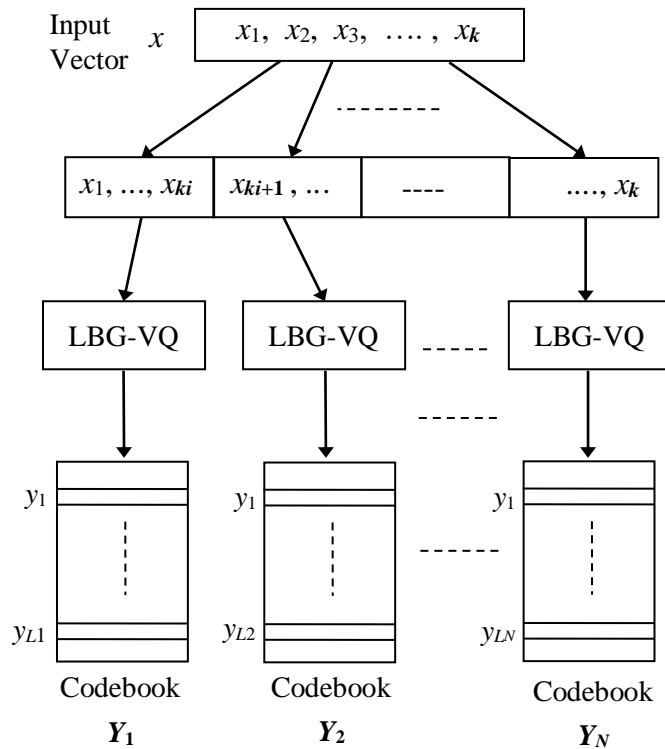


Figure 1. Bloc diagram of N -SVQ quantizer

VQ quantization of the input vector. Then, the second stage operates on the error vector between the original input vector and quantized first stage output. Practically, the quantized error vector (called residual) provides a second approximation to the original input vector leading to a more accurate representation of the input. A third stage may then be used to quantize the second stage error vector to provide further accuracy and so on. The final quantized version of the input vector is obtained by summing the output codevectors of all stages. The coding bit rate R of an M stages MSVQ is the sum of bit rates allocated to each MSVQ stage ($R = \sum_{i=1}^M R_i = \sum_{i=1}^M \log_2 L_i$). Figure 2 presents an example of a two stages MSVQ encoder/

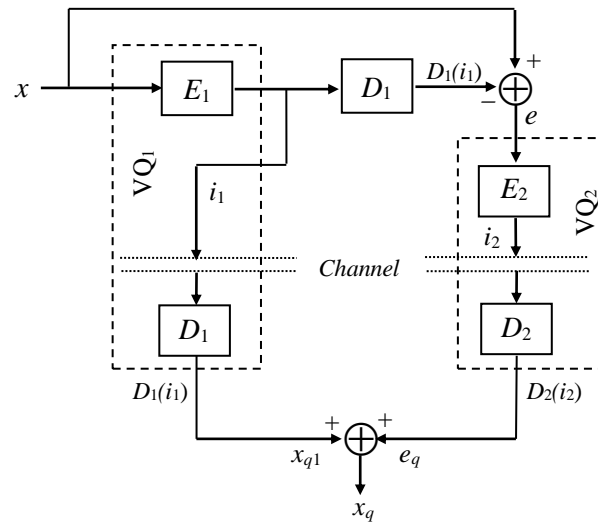


Figure 2. Two stages MSVQ encoder/decoder

decoder, where the VQ_1 of the MSVQ first stage includes the pair of the encoder E_1 and the decoder D_1 . The MSVQ quantized version of the input vector x is given by: $x_q = D_1(i_1) + D_2(i_2) = x_{q1} + e_q$.

2.2. Dither-like data hiding

Dithered quantization is a well-known technique used to reduce or to eliminate the statistical dependence between the original signal and quantization error. This is most often achieved by adding (pseudo-) random noise signal (called dither signal) to the original input signal prior quantization [11], [15].

In subtractive dithering, the dither signal is further subtracted from the quantizer's output. Thus, the total quantization error can be rendered statistically independent of the input signal as well as rendering error samples separated in time statistically independent of one another. This ensures that the power spectrum of the total error is independent of the system input, and that it is spectrally flat (white) even if the dither signal is not.

In a non-subtractive dithered system, this subtraction operation is omitted. In the subtractive dither-like data hiding (noted here SDDH) method proposed by Chang and Yu [3], the binary last stage codevector index of an MSVQ scheme are replaced with secret data bits. Thus, the last stage codevector, which is indexed now by the secret bits, is first subtracted from the original input vector to be quantized before running the MSVQ.

The SDDH idea [3] comes from the fact that in an MSVQ scheme the signals in last stages tend to be less correlated [12]. Consequently, if the codevector binary index of the last stage is replaced by the secret bits sequence to be hidden, the last stage can be viewed as a random noise that generates uncorrelated data with previous stages, which is the same as subtractive dithering [11]. By subtracting this noise data from the input of the MSVQ encoder, and adding it back at the MSVQ decoder, the degradation caused by hiding secret data can be reduced compared to the traditional non-subtractive dither (noted here NDDH) system. Let us note that in the case of conventional NDDH system, the secret data bits replace simply the MSVQ last stage binary index without vector subtraction at the first stage.

Examples of conventional NDDH and SDDH schemes are shown respectively in Figure 3-(a) and Figure 3-(b). The data hiding systems are applied to a two stages MSVQ with the second

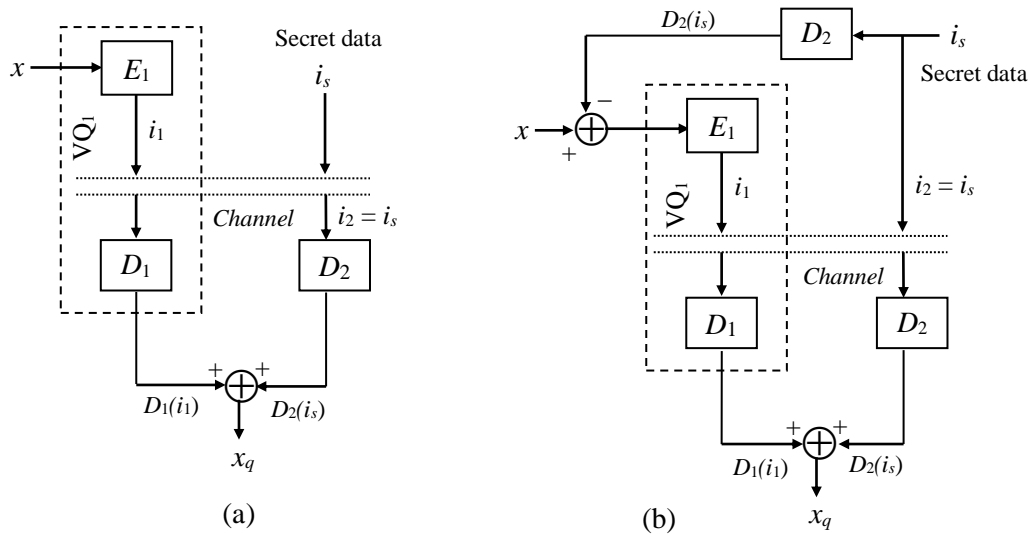


Figure 3. Two stages MSVQ dither-like data hiding: (a)- SDDH scheme, (b)- NDDH scheme

stage index used to hide secret data i_s . In the SDDH scheme, the MSVQ second stage codevector y_{is} (i.e., $D_2(i_s)$) indexed by secret sequence i_s is first extracted and subtracted from the input vector to be quantized x . Then, the first stage MSVQ is run as usual with $x - D_2(i_s)$ as input vector. The second stage MSVQ is never operated. The second codevector index i_2 , supposed to be delivered by this stage, is replaced by secret sequence i_s . Thus, the decoder receives the indices i_1 and $i_2 = i_s$ and performs exactly the same procedure as a MSVQ decoder to reconstruct the quantized version of x : $x_q = D_1(i_1) + D_2(i_s)$. At the same time, the secret sequence i_s is obtained as i_2 .

3. SPEECH STEGANOGRAPHIC SYSTEM: APPLICATION OF THE G722.2 S-MSVQ DATA HIDING SCHEME

In this section, we present a steganographic AMR-WB (G.722.2) speech coding system developed according to the S-MSVQ based data hiding method. Its basic principle consist in modifying the mechanism of the second stage of the S-MSVQ of G.722.2 ISF parameters to embed a secret speech coded by the 2.4 kbits/s MELP coder. Before presenting our speech steganographic system, let us first review briefly the S-MSVQ scheme principle.

3.1. Split MSVQ

The S-MSVQ is a hybrid scheme based on a MSVQ scheme combined with SVQ. Indeed, the S-MSVQ is a modified MSVQ with N -SVQ stages. It is structured in several successive stages, where each stage is represented by an N -SVQ instead of a simple VQ as in the conventional MSVQ. An example of a two stages S-MSVQ scheme is given in Figure 4.

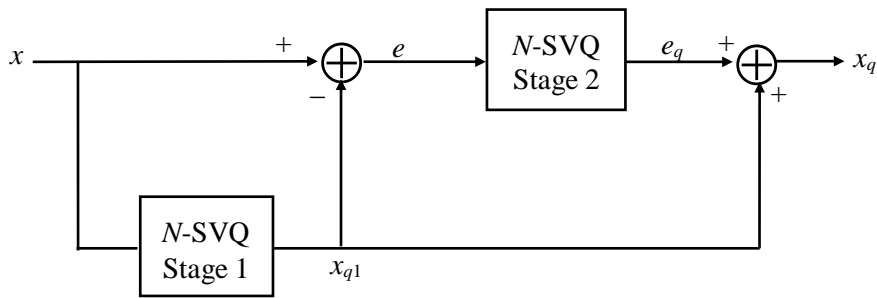


Figure 4. Two stages S-MSVQ scheme

The input vector x is first quantized by the N -SVQ of the first stage. Then, the quantization error (residual) e is used as an input to the second stage N -SVQ to obtain the quantized version e_q of the first stage residual error e . The final quantized version of x is simply the sum of the two output codevectors x_{q1} and e_q . Notice that the total number of bits allocated for quantization is divided among the S-MSVQ stages and the N split regions of each stage.

3.2. G.722.2 S-MSVQ-based data hiding scheme

Recall that the G.722.2 ISF parameters are quantized by a two stages S-MSVQ with 1st order MA predictor [7]. The standard G.722.2 S-MSVQ uses seven VQ codebooks, where two codebooks at the first stage (named here CB_{11} and CB_{12}) and five codebooks (named CB_{21} , CB_{22} , CB_{23} , CB_{24} , CB_{25}) at the second stage. Notice that the G.722.2 S-MSVQ works at 36 bits/frame for the lowest bit rate mode 0 (6.6 Kbits/s); and at 46 bits/frame for the other eight higher bit rate modes (8.85 to 23.85 Kbits/s).

For the standard 46 bits/frame S-MSVQ, 16-dimensional residual ISF vector $f_r = (f_r^1, \dots, f_r^{16})$ is split into two subvectors of dimension 9 ($f_{r1} = (f_r^1, \dots, f_r^9)$) and 7 ($f_{r2} = (f_r^{10}, \dots, f_r^{16})$), respectively. The 2 subvectors are then quantized in two stages. In the first stage, each subvector is quantized using 8 bits. In the second stage, the two quantization error subvectors $e_1 = f_{r1} - \hat{f}_{r1}$ and $e_2 = f_{r2} - \hat{f}_{r2}$ are split respectively into 3 and 2 subvectors according to the part divisions (3-3-3) and (3-4). The bit allocation for each subvector in the second stage are (6, 7, 7) bits and (5, 5) bits, respectively.

In our speech steganographic G.722.2 S-MSVQ-based data hiding system developed according to the SDDH principle, the codevectors binary indices of anyone one of the five second stage codebooks can be used to hide the secret bits sequences. The binary indices of the last stage are replaced simply by the secret bits sequence to be hidden. For a comparative evaluation, we developed also a speech steganographic G.722.2 system based on the NDDH approach.

It is important to note that in our steganographic G.722.2 systems, we can use a combination of more than one second stage codebook to perform the data hiding. Thus, the five codebooks CB_{21} , CB_{22} , CB_{23} , CB_{24} , CB_{25} can all be used in hiding process. Thus, some (or all) of the bit rate allocated to the second stage G.722.2 S-MSVQ can be used to embed secret bits. Figure 5 present an example of speech steganographic G.722.2 system where the second stage S-MSVQ CB_{25} is used for hiding secret bits sequence i_s . Notice that the bloc VQ_j in the S-MSVQ includes the pair of the encoder E_j and the decoder D_j .

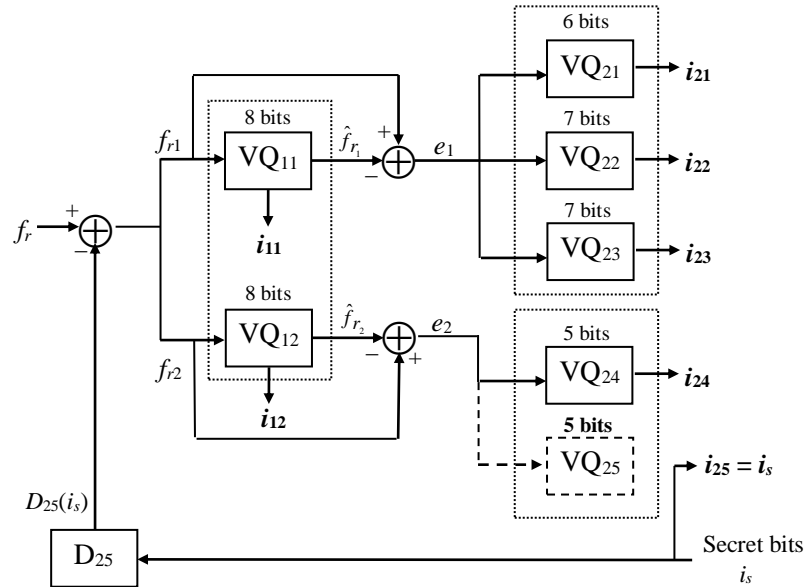


Figure 5. Example of steganographic G.722.2 S-MSVQ where the VQ CB₂₅ is used for hiding

4. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our steganographic G.722.2 speech coding systems, designed based on modifying the mechanism of the second stage of the G.722.2 S-MSVQ quantization of ISF parameters (ISFs). The data hiding S-MSVQ modification concept was carried out according to the basic idea of SDDH and NDDH approaches. The steganographic systems were called respectively S-MSVQ-SDDH and S-MSVQ-NDDH.

In our applications, the main purpose is to hide a secret speech signal coded by the 2.4 kbps MELP into a host public speech coded by the G.722.2. Notice that in all simulations, we used the G.722.2 in mode 12.65 kbits/s where the ISFs are coded by an S-MSVQ of 46 bits/frame.

4.1. Performance evaluation criteria

Performance evaluation of the implemented speech steganographic systems will be done according to the hiding capacity represented by the embedding rate of the secret speech and to the transparency (imperceptibility) represented by the perceptual quality of the speech stego-signal synthesized by the G.722.2 with embedding procedure.

The total embedding rate is given by the ratio of the number of hidden secret bits and the length of the host speech coder frame (i.e., 20 ms in G.722.2). Let us note that in our steganographic systems, the embedding rate is variable according to the combination of codebooks used in the data hiding process. Table 1 gives the embedding rates (in bits/frame and in bits/s) when using each second stage S-MSVQ codebook individually.

Table 1. Embedding rates of steganographic S-MSVQ systems

Used Codebooks	Embedding rate (bits/frame)	Embedding rate (bits/s)
CB ₂₁	6 bits	300
CB ₂₂	7 bits	350
CB ₂₃	7 bits	350
CB ₂₄	5 bits	250
CB ₂₅	5 bits	250

It should be noted that the real total embedding rate is the sum of the individual embedding rates of the codebooks used in combination in the embedding process. If we use, for example, the binary indices of the VQ codebooks CB₂₁ and CB₂₂ to hide the secret bits sequence, the embedding rate is then equal to 650 bits/s. Thus, according to the possible codebook combinations, we can use several embedding bit rates ranging from 250 bits/s (minimum embedding rate) to 1500 bits/s (maximum embedding rate).

On the other hand, for imperceptibility, we use the ITU-T Rec. P.862.2 known under the abbreviation WB-PESQ (Wide Band extension of Perceptual Evaluation Speech Quality) [16] to evaluate the coded cover/stego speech signals quality. The hidden speech signal is imperceptible if a listener is unable to distinguish between the cover and the stego speech signals; which means that the WB-PESQ difference between the two cover/stego signals is negligible.

The performance of the steganographic S-MSVQ quantizer will be also evaluated by the well-know average spectral distortion (SD) measure. The spectral distortion of each frame i is given, in decibels, by [13], [17]:

$$SD_i = \sqrt{\frac{1}{n_1 - n_0} \sum_{n=n_0}^{n_1-1} \left[10 \log_{10} \frac{S(e^{j2\pi n/N})}{\hat{S}(e^{j2\pi n/N})} \right]^2}, \quad (1)$$

where $S(e^{j2\pi n/N})$ and $\hat{S}(e^{j2\pi n/N})$ are respectively the original and quantized power spectra of the LPC synthesis filter, associated with the i^{th} frame of speech signal.

Generally, we can get transparent quantization quality if we maintain the three following conditions [13]: 1)- The average spectral distortion (SD) is about 1 dB, 2)- No Outliers frames with SD greater than 4 dB, 3)- The percentage of Outlier frames having SD within the range of 2-4 dB must be less than 2%.

4.2. Performances of steganographic S-MSVQ coding systems

For each embedding rate, we performed an optimization procedure of our steganographic systems. It consists in finding the best choice of second stage S-MSVQ codebooks that can be combined in the hiding process to obtain the best possible performance.

The speech database used in the experiments consists of 60 minutes of speech taken from the international TIMIT database ($f_s = 16$ kHz) [18]. To construct the ISF database, we used the same LPC analysis function of the G.722.2, where a 16-order LPC analysis, based on the autocorrelation method, is performed every analysis frame of 20 ms. Thus, a database of 180000 ISF vectors was constructed.

For embedding rates varying between 5 and 30 bits/frame, the SD performances of speech steganographic G.722.2 S-MSVQ-SDDH and S-MSVQ-NDDH coding systems are shown in Table 2, where the secret bits sequences are generated randomly.

For a given embedding rate, the VQ codebooks noted in the table are only the second stage codebooks (CB₂₁, CB₂₂, CB₂₃, CB₂₄, CB₂₅) in which "1" means that the corresponding codebook is used in the embedding procedure. The bit rate of this VQ codebook is then reserved for the secret bits sequence to be hidden. For example, the notation "18 (1-0-1-0-1)" means that for an embedding rate of 18 bits/frame, the codebooks CB₂₁, CB₂₃ and CB₂₅ of the modified S-MSVQ second stage were selected as best choice to be used in hiding 18 bits per each frame.

Table 2. Performance of steganographic S-MSVQ-SDDH and S-MSVQ-NDDH systems

Embedding rate (Bits/frame)	S-MSVQ-NDDH systems			S-MSVQ-SDDH systems		
	Av. SD (dB)	Outliers (in %)		Av. SD (dB)	Outliers (in %)	
		2 - 4 dB	> 4 dB		2 - 4 dB	> 4 dB
5 (0-0-0-0-1)	1.65	21.63	0.08	1.48	14.82	0.06
6 (1-0-0-0-0)	2.34	60.79	3.20	2.03	46.36	1.64
7 (0-1-0-0-0)	1.92	39.39	1.11	1.79	31.98	0.64
10 (0-0-0-1-1)	2.20	58.38	0.78	1.92	39.05	0.40
11 (1-0-0-0-1)	2.72	75.34	6.40	2.35	62.14	2.70
12 (0-0-1-0-1)	2.39	65.97	2.53	2.13	51.38	1.37
14 (0-1-1-0-0)	2.61	71.19	5.66	2.40	64.21	3.31
16 (1-0-0-1-1)	3.09	80.70	12.65	2.67	73.41	5.93
17 (0-1-0-1-1)	2.81	81.93	6.35	2.48	69.77	3.20
18 (1-0-1-0-1)	3.27	76.25	18.82	2.84	75.79	9.05
20 (1-1-1-0-0)	3.41	71.99	24.02	3.10	76.91	14.50
23 (1-1-0-1-1)	3.57	69.77	28.64	3.12	77.40	14.95
24 (0-1-1-1-1)	3.29	79.05	17.93	2.95	80.70	9.85
25 (1-1-1-0-1)	3.67	65.67	33.00	3.29	76.03	19.30
30 (1-1-1-1-1)	3.98	53.12	46.57	3.54	69.37	28.53

These results show that the SD performance degradation due to embedding process is not proportional to embedding rate. For example, for an embedding rate of 10 bits/frame, the SD degradation caused by hiding in the last second stage codebooks CB₂₄ and CB₂₅ binary indices is less than that caused by hiding in the first codebook CB₂₁ binary indices of the 6 bits/frame case. Indeed, the degradation is rather related to the importance of the used codebook in frequency domain. Knowing that the human auditory system (HAS) is more sensitive in low frequencies bands, therefore the codebooks which represent the high frequencies are less important than those of the low frequencies.

On the other hand, these SD comparative results show that the steganographic S-MSVQ-SDDH system outperform the S-MSVQ-NDDH coding system.

4.3. Performance evaluation of speech steganographic G.722.2 with ISFs quantized by modified S-MSVQ

The cover public speech database used in the following evaluations is composed of 10 speech sequences of 32s extracted from the same TIMIT database. The secret bit stream was generated by the 2.4 kbps MELP from a speech sequence of $f_s = 8$ kHz extracted from a phonetically balanced Arabic speech database [19].

Table 3 presents WB-PESQ performance comparative evaluation of the global G.722.2 where its ISF parameters were quantized by the 46 bits/frame steganographic S-MSVQ in which the second stage structure is modified according to the basic concept of SDDH and NDDH, respectively. Notice that an embedding rate of 0 bits/frame means the original standard G.722.2 without steganography (i.e. assessment of the cover speech signal). Note also that for each embedding rate, the best choices of the used steganographic second stage S-MSVQ codebooks are the same as those mentioned in Table 2.

Table 3. Performance of the global speech steganographic G.722.2 coding system

Embedding rate (Bits/frame)	G.722.2 with S-MSVQ-NDDH	G.722.2 with S-MSVQ-SDDH
	WB-PESQ	WB-PESQ
0	3.790	3.790
5	3.775	3.738
6	3.265	3.321
7	3.684	3.728
10	3.651	3.651
11	3.233	3.337
12	3.665	3.700
14	3.556	3.568
16	3.210	3.279
17	3.579	3.574
18	3.224	3.312
20	3.054	3.110
23	3.093	3.161
24	3.480	3.553
25	3.072	3.105
30	3.033	3.139

These simulation results show that for some embedding rates (5, 7, 10, 12, 14, 17 and even 24 bits/frame) the overall quality of stego-speech is almost identical to quality of cover public speech; which means that developed steganographic S-MSVQ-SDDH and S-MSVQ-NDDH techniques are practically imperceptible. Most WB-PESQ scores of the stego-signals are higher than 3.55. Hence, a good speech quality was obtained and no perceptual degradation was caused by the embedding process.

On the other hand, steganographic S-MSVQ-SDDH system yields slight improvement to the G.722.2 WB-PESQ performance compared to steganographic S-MSVQ-NDDH system.

5. CONCLUSIONS

In this paper, we developed a steganographic S-MSVQ quantizer for G.722.2 secure speech communication system. The embedding process of secret bits was carried out into the second stage S-MSVQ indices of G.722.2 ISFs according to the basic idea of subtractive (non-subtractive) dither-like data hiding. The global steganographic speech coding system was then based on embedding MELP coded secret speech into host public speech coded by the AMR-WB (ITU-T G.722.2) speech coder.

The simulation results showed that when the G.722.2 second stage S-MSVQ sub-codebooks of high frequencies bands are involved in the embedding process, our steganographic S-MSVQ-

SDDH and S-MSVQ-NDDH systems are practically imperceptible. Indeed, for some embedding rates (5, 7, 10, 12, 14, 17 and even 24 bits/frame), the G.722.2 (with S-MSVQ-SDDH) can generate stego-speech signals with similar quality to cover speech signals. Hence, the developed steganographic S-MSVQ-SDDH system can ensure a good transparency with a maximal embedding rate of 24 bits/frame (1200 bits/s). On the other hand, we can reach a maximum embedding capacity of 1500 bits/s but with a significant degradation in terms of SD and WB-PESQ. Robustness against intentional and non-intentional attacks has not been investigated in this work; it will be studied in future work.

REFERENCES

- [1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, T. Kalker. *Digital Watermarking and Steganography*, Second Edition, Morgan Kaufmann Publishers, USA, 2008.
- [2] F. Djebbar, B. Ayad, K. A. Meraim, H. Hamam, "Comparative study of digital audio steganography techniques," *EURASIP Journal on Audio, Speech, and Music Processing*, Springer, vol. 25, pp. 1-16. 2012.
- [3] P. C. Chang, H. M. Yu, "Dither-like data hiding in multistage vector quantization of MELP and G.729 speech coding," *Thirty-Sixth Asilomar Conf. on Signals, Systems and Computers*, Monterey, CA, vol. 2, 2002, pp. 1199–1203.
- [4] B. Geiser, P. Vary, "High rate data hiding in ACELP speech codecs," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2008)*, Las Vegas, Nevada, USA, March 30-April 4, pp. 4005-4008.
- [5] B. Laskar, M., Bouzid, "Vector quantization based steganography for secure speech communication system," in *Proc. 14th International Conference on Security and Cryptography (SECURITY 2017)*, vol. 4, 24-26 July 2017, Madrid, Spain, pp. 407-412. Available: <https://www.scitepress.org/Papers/2017/63983/63983.pdf>
- [6] J. He, J. Chen, S. Xiao, X. Huang, and S. Tang, "A Novel AMR-WB Speech Steganography Based on Diameter-Neighbor Codebook Partition," *Security and Communication Networks*, vol. 2018. DOI:10.1155/2018/7080673, 2018.
- [7] B. Bessette, R. Salami, R. Lefebvre, M. Jelínek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, K. Järvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, pp. 620-636, 2002.
- [8] ITU-T Recommendation G.722.2. Wideband coding of speech at around 16 kb/s using Adaptive Multi-rate Wideband (AMR-WB), 2003.
- [9] A. McCree, K. Truong, E. B. George, T. P. Barnwell, V. Viswanathan, "A 2.4 kbits/s MELP Coder Candidate for the New U.S. Federal Standard," in *Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP'96)*, 1996, pp. 200-203.
- [10] P. Moulin, R. Koetter, "Data-Hiding Codes," in *Proceedings of The IEEE*, vol. 93, pp. 2083-2126, 2005.
- [11] B. Chen, G. W. Wornell, "Quantization index modulation methods: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.
- [12] A. Gersho, R. M. Gray, *Vector quantization and Signal compression*, Kluwer Acad. Publishers, USA, 1992.

- [13] K. K. Paliwal, B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3-14, 1993.
- [14] B. H. Juang, A. H. Gray, "Multiple Stage Vector Quantization for Speech Coding," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'1982)*, Paris, France, 1982, pp. 597-600.
- [15] S. P. Lipshitz, R. A. Wannamaker, J. Vanderkooy, "Quantization and Dither: A Theoretical Survey," *J. Audio Eng. Soc.*, vol. 40, no.5, pp.355-375, May 1992.
- [16] ITU-T Recommendation P.862.2. Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs, Geneva, 2005.
- [17] S. Cheraitia, M. Bouzid, "Robust coding of wideband speech immittance spectral frequencies," *Speech Communication*, Elsevier, vol. 65, pp. 94-108, July 2014.
- [18] J. S. Garofolo et al., *DARPA TIMIT Acoustic-phonetic Continuous Speech Database*. National Institute of Standards and Technology (NIST), Gaithersburg, October 1988.
- [19] M. Boudraa, B. Boudraa, B. Guerin, "Mise en place de phrases arabes phonétiquement équilibrées," in *Proc. of XIX^{èmes} Journées d'Etude sur la Parole (JEP'92)*, Bruxelles, 1992.

LANE DETECTION FOR PROTOTYPE AUTONOMOUS VEHICLE

Sertap Kamçı, Dogukan Aksu*, Muhammed Ali Aydin

Computer Engineering Department, Istanbul University-Cerrahpasa, Istanbul,
Turkey

ABSTRACT

Unmanned vehicle technologies are an area of great interest in theory and practice today. These technologies have advanced considerably after the first applications have been implemented and cause a rapid change in human life. Autonomous vehicles are also a big part of these technologies. The most important action of a driver has to do is to follow the lanes on the way to the destination. By using image processing and artificial intelligence techniques, an autonomous vehicle can move successfully without a driver help. They can go from the initial point to the specified target by applying pre-defined rules. There are also rules for proper tracking of the lanes. Many accidents are caused due to insufficient follow-up of the lanes and non-compliance with these rules. The majority of these accidents also result in injury and death.

In this paper, we present an autonomous vehicle prototype that follows lanes via image processing techniques, which are a major part of autonomous vehicle technology. Autonomous movement capability is provided by using some image processing algorithms such as canny edge detection, Sobel filter, etc. We implemented and tested these algorithms on the vehicle. The vehicle detected and followed the determined lanes. By that way, it went to the destination successfully.

KEYWORDS

Autonomous Vehicle, Lane Detection, Image Processing, HSV Color, RGB Color, Canny Edge Detection, DC Motor, Region of Interest (ROI), Vanishing Point, Sobel Filter

1. INTRODUCTION

Lane tracking is one of the most important tasks for autonomous vehicles. There are one or more lanes on each road. In a vehicle without lane tracking, there is no steering and this causes to accidents.

Prior to the development of autonomous vehicles, the vehicles had Active Lane Tracking Assistance. This system is used by many famous vehicle brands due to the lack of autonomous vehicles. The application of this system includes different types. These are just sound and vibration warning, steering interventions instead of the driver etc. This system helps keep the vehicle in the lane. Therefore, the importance of lane tracking system is understood.

A driving experience without lane tracking is not considered. In order to move with the correct timing, it must follow the path in real motion while in motion. This monitoring is usually performed in accordance with the timing of the images taken from the camera.

The algorithm of the vehicle allows to protect the lane. It adjusts the vehicle speed according to the lane. In addition, with the designed algorithm, the autonomous vehicle gives the

vehicle an idea of what kind of moves it should follow if it wishes to cross a different lane in the future. First, the lane where the autonomous vehicle is located has to be detected and the snapshots are captured from the camera. These snapshots are called frames. There are many methods of image processing and are used to detect the lane. The lane lines are with the help of the camera which can see both sides of the vehicle. First, the quality of the picture should be improved and the lines should be determined. After that, the vanishing point and the Region of Interest (ROI) are determined. With the determined ROI, the vehicle understands how the lane continues in the direction. With the determination of the lane, the vehicle is given the instructions to continue straight to the vehicle according to the angles determined by the lane and to turn right or left.

This paper is organized as follows: Section 2 contains relevant studies and provides a comprehensive overview of the different detection methods that are used to the project. The functions and models used in Chapter 3 are presented. The experience gained in Chapter 4 is explained and the project output is given. In Chapter 5, the operating logic of the car following the lane is explained with a flow diagram. The results obtained are given in Chapter 6.

2. RELATED WORKS

The interest in autonomous vehicles has been increasing rapidly in recent years. In this paper, image processing algorithms for lane detection and tracking are mentioned.

There are several techniques for performing lane detection. First of all, the image processing methods used in the project should be examined to find the lines. Thereafter, lines should be identified from image frames where image processing techniques are applied. After this step, the ROI between the lines should be found. In addition, a virtual line is drawn in the middle of these lines to make the path look like a curve. With this curve, the bend of the way is determined by finding the angle. With this angle, the vehicle has an angle of movement and the vehicle moves in a straight, right-handed, left-handed direction according to the specified angle.

In this paper, we examine the related studies in three stages. These steps are: image processing techniques, detecting lane lines and finding ROI.

2.1. Image Processing Techniques

In the design of the algorithm, firstly the mutation from RGB color space to HSV color space is made. According to some studies, it is more convenient to use HSV color space when we want to differentiate an object of a certain color in any computer vision/image processing application [1]. In addition, the HSV color space provides clarity of the colors in the image. HUE is also used to distinguish colors more clearly.

Color images contain more information than gray level images. For this reason, in some related studies, the edge detection process for color images has been examined. Grayscale method is one of these methods. The cost of detecting ROI in the image was reduced by the Grayscale image conversation method and the process was accelerated [2,3].

2.2. Determination of Lane Lines

Canny edge detection, Hough transformation and Sobel filter methods are the methods used to find the lane lines.

Canny edge detection is a method of edge detection. The lanes are lines and have edges. Hough transformation method is a method that finds and shows shapes. Since the lane lines have a shape, lane lines can be found by the Hough transform method [4].

Sobel filter method is a separate method used to find the edge.

It is seen in all the studies obtained the use of the edge detection algorithm in images with a grayscale colour space is closer to the edge information in the actual image.

2.3. Finding Region of Interest (ROI)

The area of the lane is called the ROI. While some algorithms have an ROI up to the point where the lines can be detected by combining the lane lines in the image, some algorithms have methods for finding the ROI based on vanishing point detection techniques [5].

There are two common methods for the detection of ROI. Firstly, left and right lane lines are found. In the first method, the lane lines are stretched and intersect at one point. A triangular region is formed. The area within the triangular region up to the region where the lane are detected forms a rectangular region. This field returns the ROI. The deviation difference of the angles and the obtained region is calculated and the direction of the path is calculated. [1]

Another method is to determine the ROI by determining the skyline. Horizon line is the line where the earth globe and sky intersect when viewed in nature. With the determination of the horizon line, the lane lines are also combined and the intersecting area shows the ROI.

3. DESIGN OF THE LANE FOLLOWING AUTONOMOUS CAR

The design part of the car is divided into two parts. These are the movement of the software and the hardware of the vehicle. The software field is algorithm design and coding.

3.1. Software

The designed code has a layered architecture. There is a python file with a main structure named Main. Methods in other Python files are imported and run by calling the methods respectively. The lane lines must be determined in the algorithm of the car following the lane. For this, lane detection algorithm is used. Image processing techniques are used in the lane detection algorithm. These image processing techniques are described in the below.

By applying image processing techniques, lane lines are determined from the obtained image. These lane lines are calculated and centered in the algorithm and center lane is formed. The algorithm written for ROI detection is called here and the calculation of the ROI is done in this algorithm. The ROI-finding algorithm maintains the intersecting and polygon definition region on the image. This is the area between the lane and the horizon. In order to detect the lanes, image processing methods are applied to the given picture frames of the frame name taken from the image in the received camera. These methods; Conversion from RGB to HSV color space, Gaussian Blur, Gray Scale method, Canny Edge Detection, Hough Transform methods were used.



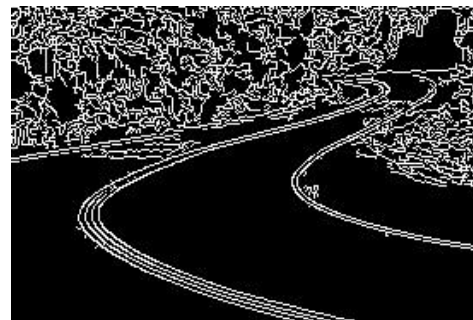
(a) Original frame



(b) HSV frame



(c) Grayscale frame



(d) Canny Edge Frame Figure 1. Steps of image processing methods on a path

In Figure 1 (a), a path with RGB color space was converted to HSV color space as in Figure 1 (b). Color saturation increases with HSV color space. Then, the image was processed as shown in Figure 1 (c) using Gaussian Blur and Gray Scale methods. Figure 1 (d) is used to understand the line of the lanes on the road by using Canny edge detection and Hough Transform methods. As a result of the calculations, the strip lines are perceived as shown. In Figure 2-a, a re-converted image frame is given to the RGB image. In Figure 2-b, an image frame that is not converted to RGB color space is given.



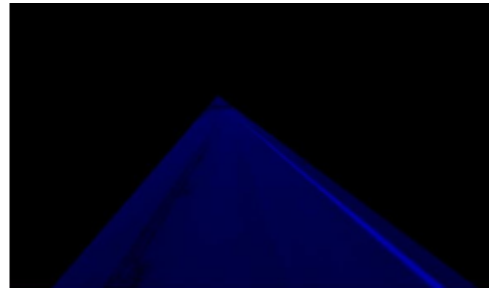
(a) Original path image



(b) Detected Lane from image



(c) Detected interested lane



(d) ROI of the path image Figure 2. Detection of lanes in a sample path

In Figure 2 (a), the algorithm is written to detect the road where the vehicle is located and this lane is shown based on the nearest lane. In Figure 2 (b), the ROI of the lane was found.

In Figure 2-c, the algorithm is written to detect the path where the vehicle is located and this lane is shown based on the nearest lane. Figure 2-d shows the ROI of the lane.

3.2. The Movement of the Vehicle

We want the vehicle to center the strips before start moving. In the code, the function that generates the virtual line is written to center. Then you need to follow the lane where it is located. For tracking of the lane, the ROI has the angle of rotation of the lane. When driving straight ahead, the vehicle is commanded to continue straight ahead. We used DC motors for the movement of the vehicle.

```
GPIO.output(Motor1A,GPIO.LOW)
```

Above left is the left motor stop command. In the same way, the right engine must be stopped for a right turn. This stop command may be slightly different, if the vehicle is not going to turn completely and rotating at a different degree than the angle of 180, the motor in the direction of rotation is decelerated and the motor in the other direction is accelerated. This is the logic of the vehicle's progress and rotation relative to the road.



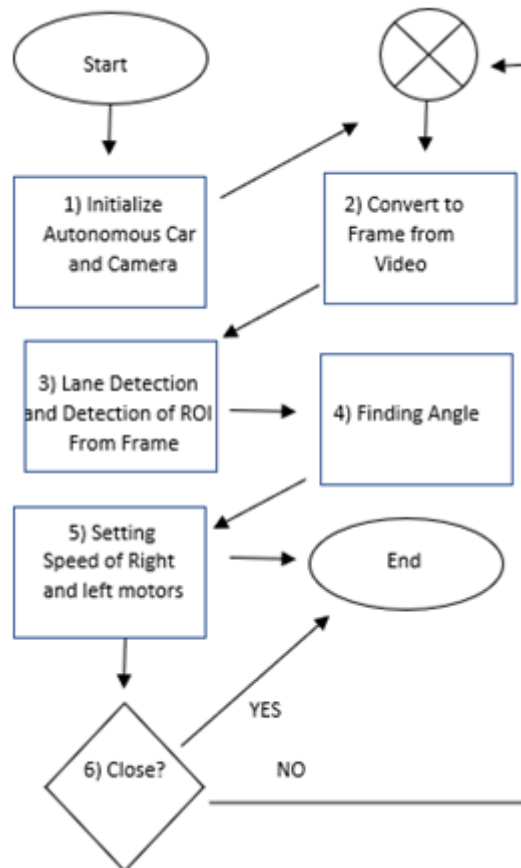
Figure 3. Image of autonomous vehicle prototype

4. EXPERIMENTS

The most important part was the progress of the coding step by step. During the last construction phase of the vehicle, the movement of the vehicle in relation to the road in a road video was observed by remote connection to the computer without connecting the camera. The code was then executed according to the images taken from the camera. In both structures, the vehicle has

been shown to proceed in a desired manner. This study was also carried out as a graduation project and successfully completed.

5. THE PROPOSED METHOD



Algorithm. Pseudocode for the proposed method

```

1: // autonomous car and camera setting
2: while !CloseCar() do
3:   convert to frame from video
4:   detect to lane and ROI from frame
5:   find to angle of load
6:   set to speed of motor
7: end while
  
```

Figure 4. The proposed method: The Flowchart and The Pseudocode.

The figure above shows the algorithm-based operation designed in the period from the start of the vehicle to the closing of the vehicle.

When the vehicle is started, the camera also operates (Scheme1) and instant frames (frames) are taken from the camera (Scheme2). Lanes are detected by image processing methods and ROI is found (Scheme3). When the ROI is found, it is understood how the vehicle should be directed to protect the lane and there is an angle for this (Scheme 4). The angle found is adjusted by the speed of the right and left motors (Figure 5). As long as the vehicle is not switched off, the camera operates and returns to Scheme2, resulting in a flow loop. If the vehicle is switched off

(Scheme6 takes yes), the flow switches to Scheme7 and the autonomous vehicle and camera are switched off.

6. CONCLUSION

In this paper, an autonomous vehicle prototype that detect lanes via image processing techniques, which are a major part of autonomous vehicle technology is presented. Some image processing algorithms such as canny edge detection, Sobel filter, etc. are used to provide autonomous movement capability. They were implemented and tested on our prototype vehicle. The prototype vehicle detected and followed the determined lanes successfully and it reached the destination. As a future work, we are planning to use generative neural networks, deep learning, and various machine learning algorithms to detect lane and traffic signs together.

ACKNOWLEDGMENT

The project in this paper is presented as a bachelor thesis and it is supported by Istanbul University-Cerrahpasa Scientific Research Projects Commission as project number 32561 and "FBA-2019-33004". We would like to thank Istanbul University-Cerrahpasa Scientific Research Projects Unit for their support.

REFERENCES

- [1] Gurjashan Singh Pannu, Mohammad Dawud Ansari, Pritha Gupta," Design and Implementation of Autonomous Car using Raspberry Pi", International Journal of Computer Application, 9 March 2015
- [2] Peerawat Mongkonyong, Chaiwat Nuthong, Supakorn Siddhichai, Masaki Yamakita, "Lane detection using Randomized Hough Transform", 8th TSME-International Conference on Mechanical Engineering, January 2018.
- [3] A.A.M. Assidiq, Othman O. Khalifa, Md. Rafiqul Islam, Sheroz Khan, "Real time lane detection for autonomous vehicles", IEEE Xplore, 2008.
- [4] Ze Wang , Weiqiang Ren "LaneNet: Real-Time Lane Detection Networks for Autonomous Driving", ArXiv, 4 July 2018
- [5] CHANG YUAN, HUI CHEN , JU LIU , DI ZHU , AND YANYAN XU, "Robust Lane Detection for Complicated Road Environment Based on Normal Map", IEEE, July 27th 2018.
- [6] Aksu D., "Performance analysis of log-based intrusion detection systems", MSc, Istanbul University, Istanbul, Turkey, 2018.
- [7] Thittaporn Ganokratanaa , Mahasak Ketcham, Sasipa Sathienpong, "Real-Time Lane Detection for Driving System Using Image Processing Based on Edge Detection and Hough Transform", International Conference on Digital Information and Communication Technology and its Applications, 2013
- [8] Shobit Sharma, Girma Tewolde, Jaerock Kwon, "Behavioral Cloning for Lateral Motion Control of Autonomous Vehicles Using Deep Learning", 2018 IEEE International Conference on Electro/Information Technology (EIT), May 2018.
- [9] David C. Andrade, Felipe Bueno, Felipe Franco, Rodrigo A. Silva, Student Member, IEEE, João H. Neme, Erick Margraf, Student Member, IEEE, William Omoto, Felipe Farinelli, Angelo M. Tusset, Sergio Okida, Max M. Santos, Senior Member, IEEE, "vel Strategy for Road Lane Detection and Tracking based on Vehicle's Forward Monocular Camera", IEEE Transactions on Intelligent Transportation Systems, September 2018.

- [10] Sertap Kamçı, Aylin Yazıcı, Bilge Kamberoğlu, Doğukan Aksu, Muhammed Ali Aydın, "A Survey of Lane Detection and Traffic Signs Recognition for Autonomous Vehicles", Akademik Bilişim'2019, Ordu Üniversitesi Bilim ve Teknoloji Dergisi, 2019.
- [11] Gurveen Kaur, Dinesh Kumar, "Lane Detection Techniques: A Review", International Journal of Computer Applications, IJCA Journal, 2015.
- [12] Anjali Goel, "Lane Detection Techniques - A Review", International Journal of Computer Science and Mobile Computing, February 2014.
- [13] C.Y. Kuo, Y.R. Lu and S.M. Yang, "On the Image Sensor Processing for Lane Detection and Control in Vehicle Lane Keeping Systems", US National Library of Medicine National Institutes of Health-NCBI, April 2019
- [14] Min Bai, Gellert Mattyus, Namdar Homayounfar, Shenlong Wang, Shrinidhi Kowshika Lakshmikanth, Raquel Urtasun, "Deep Multi-Sensor Lane Detection", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2018
- [15] Aharon bar hillel, Ronen Lerner, Dan Levi, Guy Raz, "Recent progress in road and lane detection: A survey", Machine Vision and Applications 2011.
- [16] Qingquan Li, Jian Zhou, Bijun Li, Yuan Guo, Jinsheng Xiao, "Robust La", Dec 2018 ne- Detection Method for Low-Speed Environments", MDPI, Dec 2018.
- [17] Nur Shazwani Aminuddin, Masrullizam Mat Ibrahim, Nursabillilah Mohd Ali, Syafeeza Ahmad Radzi, Wira Hidayat Mohd Saad & Abdul Majid Darsono, "A New Approach to Highway Lane Detection by Using Hough Transform Technique", 2 (Dec) 2017, Journal of IC

AUTHORS

Sertap Kamçı completed her primary school education in Incirli Bahçe Primary School. She completed her secondary school education at the Siir Mektebi Primary School and his high school education at Gungoren Anatolian High School. She graduated Istanbul University- Cerrahpasa Faculty of Engineering Computer Engineering at 2019.



Doğukan Aksu received his B.S. and M.Sc. degrees in Computer Engineering at Istanbul University in 2015 and 2018 respectively. He is a PhD student in Computer Engineering. His research interests are information and network security, cryptography, artificial intelligence, machine learning and image processing.



Muhammed Ali Aydın completed his B.S. in 1997-2001 in Computer Engineering at Istanbul University, M.Sc. in 2001-2005 in Computer Engineering at Istanbul Technical University and Ph.D. in 2005-2009 Computer Engineering at Istanbul University. His research interests are communication network protocols, network architecture, cryptography, information security and network security.



FINDING MAXIMAL LOCALIZABLE REGION IN WIRELESS SENSOR NETWORKS BY MERGING RIGID CLUSTERS

Saroja Kanchi

Department of Computer Science, Kettering University, Flint, MI, USA

ABSTRACT

Localization of Wireless Sensor Network (WSN) is the problem of finding the geo-locations of sensors in a sensor network deployed in various applications. Given the proliferation of sensors in various applications, the localization and tracking of sensors have received considerable attention. Properties of rigidity and flexibility of the underlying graph of the WSN have been studied as a means of determining the localizability of the nodes in the WSN. In this paper, we present a new 3-merge technique for merging three rigid clusters of a network graph, into larger rigid cluster and we use this algorithm for finding maximal localizable regions within the WSN. We provide simulation results on random deployments of WSN to prove that this technique outperforms previously known algorithms for finding maximal localizable subregions. Moreover, simulation results show that the number of anchors needed to localize the entire WSN decreases due to finding large localizable regions.

KEYWORDS

Wireless Sensor Network, localization, rigidity, cluster, merging

1. INTRODUCTION

Wireless sensor networks are a collection of sensor nodes deployed in various applications including environmental monitoring, search and rescue missions, autonomous driving, target tracking, healthcare monitoring, forest fire detection etc.[13][7][16][17]. Awareness of the exact location of the sensors is crucial to the success of these applications. Once deployed, in a majority of these applications, the sensors move after deployment and therefore predetermining the location of the sensors is not practical. Moreover, it is not always possible to equip the sensors with GPS due energy consumptions and obstructions in indoor applications. Determining the ge-locations of sensors is the problem of *localization* of a WSN.

There have been several approaches to localization depending the capability of the sensor nodes to obtain various information about the context it is in, and whether the algorithm is centralized or distributed[9]. The *range-based* approaches assume that sensors can find the distance to neighboring sensors within their sensor radius using RSSI signal strength, or time difference of arrival (TOA) between radio signals. In addition, angle of arrival (AOA) of a signal [10], can be used determine the location of the sensors. In range-free approaches, where distance between sensors is not known, [2][12], number of hops is used for localization. Many approaches assume that there are specialized anchors whose location is known, in cases where limited number of GPS equipped sensors are available. In range-free localization, number of hops to an anchor is used as a means of locating sensors.

In self-localization, nodes localize using distributed computation [9][11], by exchanging information with surrounding nodes. The geometric property of location of nodes dictates that given a set of nodes whose locations are known and an unlocalized nodes whose distance to three localized nodes are known, the location of the unlocalized node can be uniquely determined. The process of thus growing localized set of nodes by spanning out the localized nodes is called *trilateration*. Trilateration [1],[18] is commonly used as a means of localizing nodes, and often a variation of bilateration is used to find location of nodes. Moreover localization is assisted by a mobile anchor or mobile robot that help add missing distances between sensor nodes [15][19].

In centralized approach where each node sends its data to a centralized server, the distance map or any other information provided can be used for localization. The MDS-MAP [6] technique finds the missing distances using shortest path algorithm and uses distance matrix for localization. It turns out that given a distance map between nodes, finding the exact geolocations of nodes is closely related to the problem of rigidity of the underlying network graph. Therefore finding large rigid subgraphs within a WSN is extremely useful in localizing large number of sensor nodes. Therefore there has been considerable interest in using rigidity for localization. [3][9]. Recently, Erin [4] has proposed a new graph invariant for graph rigidity, namely redundancy index and rigidity index. There exists a unique realization of the graph onto 2D plane if and only if the given the WSN is uniquely localizable. While checking if a graph is globally rigid is polynomially solvable, finding locations of the nodes in a rigid graph is NP-Complete. Using our polynomial time algorithm, large globally rigid subregions can be found in the network each of which are localizable. Note that actual realization of this lower bound would require 3 anchors per rigid region, and using MDS-MAP algorithm for each rigid region and merging the local maps to obtain a global map. Our experimental results indicate that this new 3-merge technique localizes large number of nodes in any randomly deployed WSN.

It is shown that even in sparse networks, a large percentage of nodes can be localized with as few as 3 anchor nodes. The paper is organized as follows. In Section 2, we provide the details of rigidity theory of graphs. In Section 3, we provide the new 3-merge technique used for finding. This technique extends the 2-merge technique given in [8]. In Section 4, we present the localization algorithm using the new theorem. In Section 5, we present the results of simulation.

2. GRAPH RIGIDITY AND LOCALIZATION

In this section, we introduce the theory in network localizability and rigidity. A detailed description can be found in [3].

Given a network graph WSN sensor nodes $1..n$ and distances between a subset of node pairs, the network localization problem is to determine the unique locations of the nodes such that the euclidean distance between the location of sensor node i and sensor node j is the distance between sensor nodes i and j of the sensor network.

We model the network as a graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ denote the sensor nodes of the network and an edge $(v_i, v_j) \in E$ exists if the distance between v_i and v_j is known. The edge weight w_{ij} , denotes the distance between the nodes v_i and v_j . The network localization problem is to determine the locations of V such that the euclidean distance between the vertex locations is equal to the edge weight w_{ij} , for each edge $(v_i, v_j) \in E$. If under the given constraints, there is only one position for each node, then the network is localizable. The problem of localization is to find the unique location of each vertex subject to given distance information between vertices.

The network localization problem is closely related to the Euclidean graph realization problem. A *framework* of a graph G is a mapping of vertices of G onto 2D plane, such that distance between two vertex placements precisely equal the edge weight of the corresponding edge in G . We can think of this framework as bar and joint framework, where bar corresponds to edges and joint corresponds to vertices. The bar-and-joint framework is *generically rigid* if it has only trivial deformations, as shown in Figure 1, e.g., translations and rotations. Laman characterized rigidity combinatorially [10].

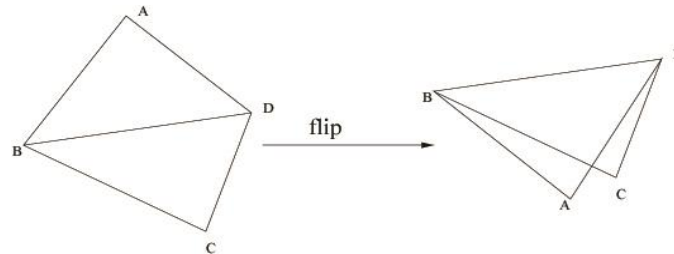


Figure 1. Generically Rigid Graph

Laman's theorem can be intuitively explained as follows. For a two dimensional graph with n vertices, the positions of its vertices have $2n$ degrees of freedom, of which three are the rigid body motions. Therefore graph is rigid if there are $2n - 3$ constraints. If each edge adds an independent constraint, then $2n - 3$ edges should be required to eliminate all nonrigid motions of the graph. Clearly, if any induced subgraph with n vertices has more than $2n - 3$ edges then these edges cannot be independent which leads the following version of Laman's theorem [10].

Theorem 1. The edges of a graph $G = (V, E)$ are independent in two dimensions if and only if no subgraph $G' = (V', E')$ has more than $2n' - 3$ edges, where n' is the number of nodes in G' .

Corollary 1. A graph with $2n-3$ edges is generically rigid in two dimensions if and only if no subgraph G' of G has more than $2n' - 3$ edges, where n' is the number of nodes in G' .

A framework (G, p) is *globally rigid* if, the distance between every pair of nodes is preserved for different framework realizations, and not just those defined by the edge set. If a graph $G = (V, E)$ is generically rigid but contains more than $2n-3$ edges, then G is called a *redundantly rigid* graph. For such a graph, $G - e$ is rigid for all $e \in E$. An edge is called a *redundant edge* if graph remains rigid after its removal. It is known that G has a unique generic realization, i.e globally rigid in 2-space if and only if G is 3-connected and redundantly rigid [22]. Therefore, in order to find unique locations of nodes in a network, we need the underlying graph to be globally rigid and vice-versa.

The problem determining localizability thus reduces to the problem of finding global rigidity. The globally rigid subregions of a graph become localizable and vice versa.

3. ALGORITHM FOR FINDING RIGID CLUSTERS

In this paper, we set out to find maximal localizable subregion within a network by finding the maximal subregions that are globally rigid. This is done by first finding small globally rigid regions within a network and annexing nearby globally rigid regions.

The algorithm we use for checking redundant rigidity is the polynomial time pebble game algorithm proposed by Jacobs[5]. The graph's 3-connectivity property is easily checked in polynomial time.

Given a graph $G = (V, E)$, we define a R_i as subregion of G if $R = (V_R, E_R)$ is globally rigid. In the theorems below, we provide techniques that merge two or three globally rigid regions into larger globally rigid regions. The following theorem [8] provides a technique called 2-merge, for merging two globally rigid regions.

Theorem 2: Given globally rigid graphs $R_1 = (V_1, E_1)$ and $R_2 = (V_2, E_2)$, the graph formed by merging the two regions, $R_{2\text{-merge}} = (V_1 \cup V_2, E_1 \cup E_2 \cup E')$ consisting of additional edges E' described in one conditions (a) to (d) is a globally rigid graph.

- a. There are three or more vertices in common between V_1 and V_2 . The additional edges consist of edges with one end point in V_1 and other in V_2 , and E' could be empty.
- b. There are two vertices in common between V_1 and V_2 , and there is at least one additional vertex in V_1 that has at least one edge connecting to a vertex V_2
- c. There is one vertex in common between V_1 and V_2 , and there are at least two other vertices that each have at least one edge connecting to a different vertex in V_2
- d. There are no vertices in common between V_1 and V_2 , and there are at least 3 vertices in V_1 ($i=1,2$) each have an edge connecting to a different vertex in V_2 ($j \neq i$) and there are at least 4 edges between vertices of V_1 and vertices of V_2 .

The proof can be found in [8].

In this paper, we provide a technique, 3-merge, for merging three globally rigid regions into a single globally rigid region.

Theorem 3: Given globally rigid graphs $R_1 = (V_1, E_1)$, $R_2 = (V_2, E_2)$ and $R_3 = (V_3, E_3)$ the graph formed by merging the three regions, $R_{3\text{-merge}} = (V_1 \cup V_2 \cup V_3, E_1 \cup E_2 \cup E_3 \cup E')$ is globally rigid if there are 7 edges connecting the three graphs in such a way that no two regions have more than 4 edges between them and there are at least 3 vertices in each region that have an edge connecting to another region.

Proof: We will prove that the graph $R_{\text{merge}} = R_1 \cup R_2 \cup R_3 \cup \{e_i, i = 1, 7\}$ is a globally rigid graph. Note that each R_i $i=1,3$ are 3-connected and redundantly rigid by Corollary 1. We will prove that the graph $R_{3\text{-merge}}$ is also 3-connected and redundantly rigid.

Since each R_i is 3-connected, there are three vertex disjoint paths between any two vertices with the same R_i , $i=1,3$. Therefore, WLOG, it is sufficient to prove that there are three vertex disjoint paths from one vertex v_i of R_1 to v_j of R_2 . Since there are 7 edges between the three regions, if there are no edges between R_1 and R_2 , there must be at least 4 edges between R_1 to R_3 or R_2 to R_3 and this is not the case. Therefore, there is at one edge between R_1 and R_2 .

Also, note that there are three vertices in R_1 which have edges with the other endpoint in R_2 or R_3 . If three of these edges are between R_1 and R_2 , then we have three vertex disjoint paths between any vertex of R_1 and any vertex of R_2 . If there are two vertex disjoint paths using direct edges, then there must be a third vertex in R_1 , connects to R_3 . Since there are three edges from R_3 to R_2 , (since otherwise the total number of edges between three regions will be less than 7) ,we can use one of the paths from R_2 to R_3 , to find the third path from a vertex in R_1 and a vertex in R_2 . Thus proving 3-connectivity of $R_{3\text{-merge}}$

To prove redundant rigidity, we will show that removal of any edge leaves the graph generically rigid. Clearly, each graph R_1 , R_2 , and R_3 is redundantly rigid, which means that removing any edge from any of the graphs, the remaining graph contains $2n_1-3$, $2n_2-3$ and $2n_3-3$ edges spanning the n_1 , n_2 and n_3 vertices such that each of these are independent edges in R_1 , R_2 and R_3 respectively. We will prove that removing an edge from $R_{3\text{-merge}}$, still leaves an independent

set edges of size $2(n_1+n_2+n_3) - 3$ edges. Removing any one of the 7 cross edges, the remaining graph contains $2(n_1+n_2+n_3) - 9 + 6 = 2(n_1+n_2+n_3) - 3$ edges. We will prove that the graph containing the independent edges from each region and the six cross edges, forms an independent set of edges for the merged region.

Note that by Theorem 2, a graph G with n vertices and $2n-3$ edges is an independent graph (i.e. all of its edges are independent) if there is no subgraph of G , of k vertices with more than $2k-3$ edges. Let us consider subgraph S of $R_{3\text{-merge}} - \{e\}$ where e is any cross edge. If the S contains no cross edges, then S is independent due Corollary 1. Consider a subgraph that contains all of the six cross edges. Any subgraph that includes all of these 6 edges, there no more than $2_{k_1}-3$, $2_{k_2}-3$ and $2_{k_3}-3$ in each of the subgraphs. Therefore, there are no more than $2(k_1+k_2+k_3)-9+6 = 2k-3$ edges in the subgraph that includes all of the cross edges, proving redundant rigidity. For a subgraph that includes less than the maximum number of cross edges, the same argument holds

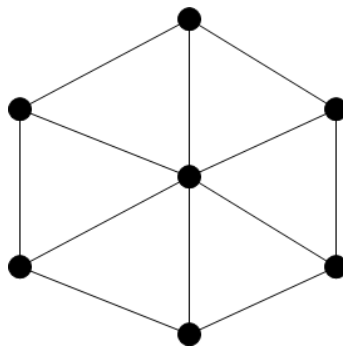


Figure 2. A Sample Wheel Graph

4. LOCALIZATION ALGORITHM USING RIGID CLUSTERS

Given a WSN graph we perform a centralized algorithm as follows:

1. Find the one-hop globally rigid regions by considering the one-hop neighbors of each vertex. These have wheel structures and they might have vertices in common. See Figure 2.

Repeat steps i) to iii) until no additional merges are possible.

- i) To merge two one-hop rigid regions, we use 2-merge of Theorem 2 with three common vertices we find one the conditions of (a). This step is repeated until no additional 2-merges are possible,
 - ii) The resulting rigid regions from Step i) are merged using 2-merge with edges in common by looking to see if conditions (b), (c) and (d) of Theorem 2 hold. Again this step is repeated until no additional merges are possible.
 - iii) The resulting rigid regions from Step ii) are merged using 3-merge algorithm of Theorem 3, looking for three regions for which conditions of 3-merge hold. This step is repeated until no additional merges are possible.
2. Once the large rigid regions are thus formed, we use 3 anchors in each rigid region to localize the rigid regions.

5. SIMULATION RESULTS

Simulation was performed on Matlab, using 100 nodes on a 100 by 100 square foot area with various radii from 12 to 22. The nodes were uniformly distributed over the area. The results in

Figure 3, show that even for really sparse networks with radius as low as 17, significant number of nodes belong to rigid regions and can be localized using three anchors per region.

Figure 3 shows that using 3-merge the number of anchors needed for localizing all localizable nodes dramatically decreases when the radius goes from 12 to 22 and 3 anchors suffice for localizing more than 98% of the nodes when the radius is 22, as indicated in Figure 4. Figure 5 demonstrates that this technique finds really large rigid subgraph and largest rigid subgraph size contains most of the nodes for networks of radius 22. Figures 6 and 7 demonstrate that even for a sparse graph, the number of rigid regions that the algorithm finds are numerous as outlined by the cyan edges. Figures 8 and 9 demonstrate that for dense graph with a radius of 22, all rigid regions are merged into single rigid region making the entire network localizable with just 3 anchors.

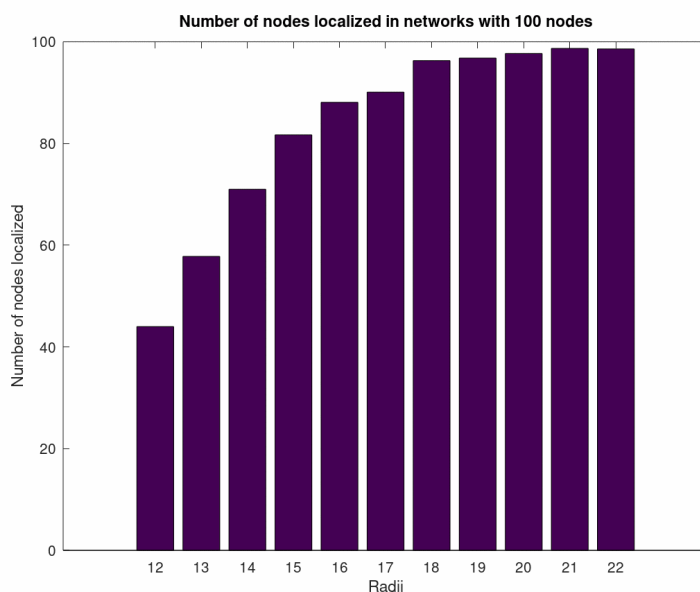


Figure 3: Number of nodes in rigid regions with 3-merge

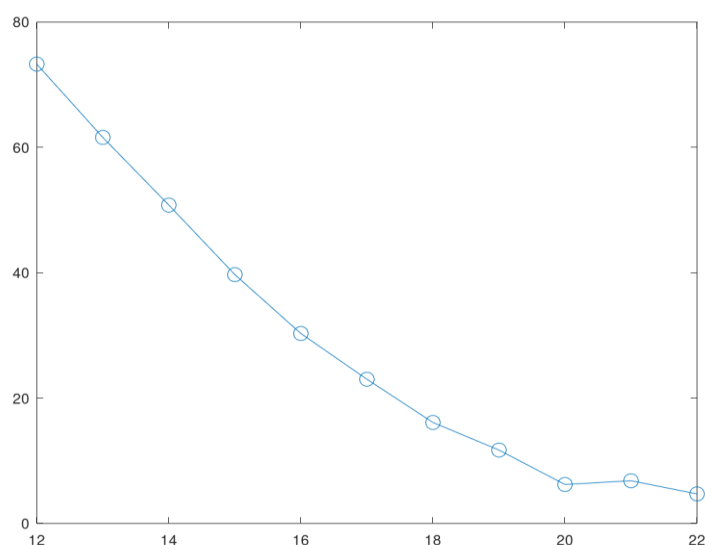


Figure 4: Number of anchors needed to localize the nodes in rigid regions

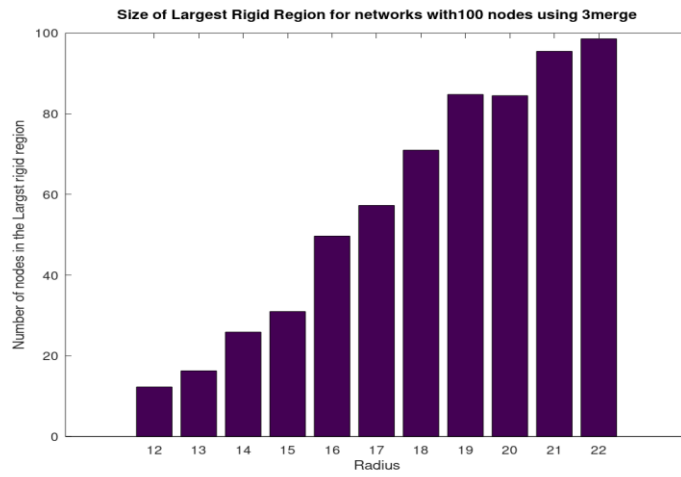


Figure 5: Size of largest rigid region when network regions merged using 3-merge

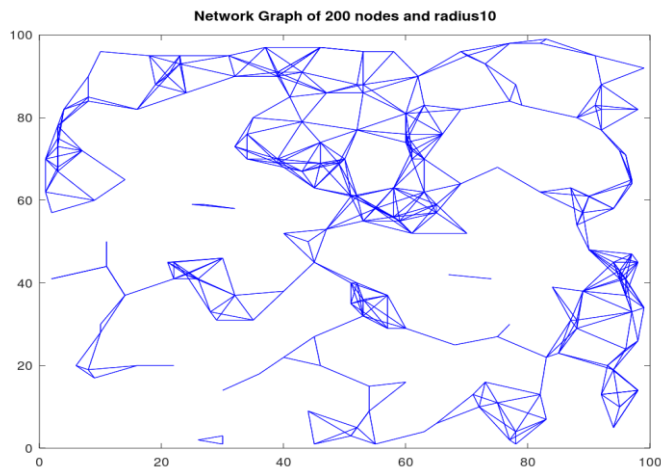


Figure 6: A sparse network graph with radius of 10

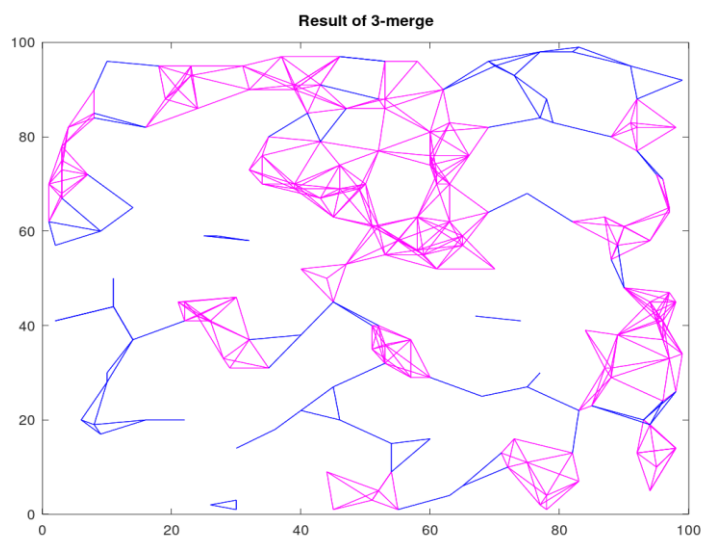


Figure 7: Rigid regions found in the graph in Figure 4 using 3-merge algorithm

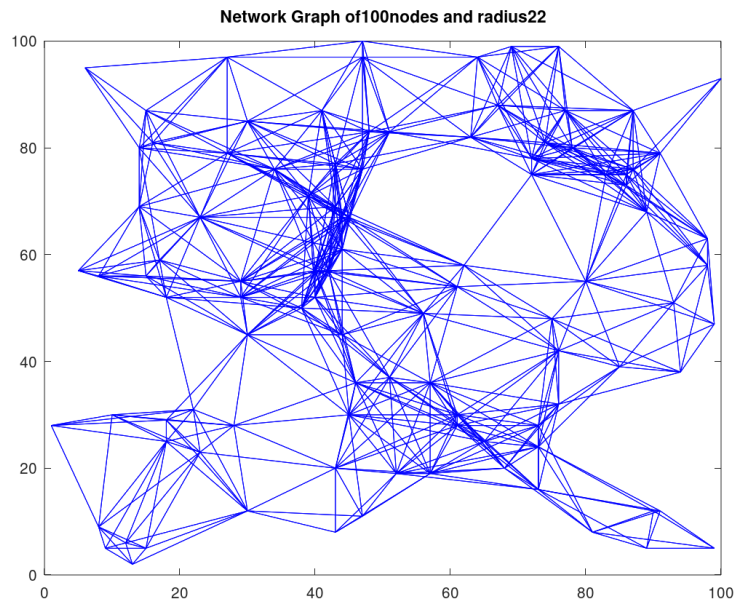


Figure 8: Network with 100 nodes and radius of 22

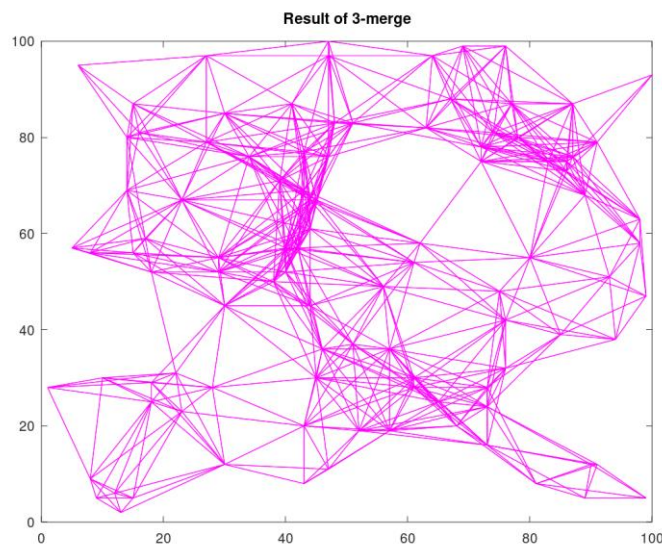


Figure 9: Rigid regions found in the graph in Figure 6 using 3-merge algorithm

6. CONCLUSION

The paper presents a new theorem for merging two rigid regions into a single rigid region for a graph. This theorem is used to find large rigid regions in a network graph, starting with wheel graphs and merging them using 2 merge algorithm first and then merging three regions at a time that obey the conditions of the theorem. The simulation results show that when the radius of the network graph is 19 feet or above in a 100 by 100 feet² network the number of nodes localized is over 80%. When the network is sparse, it is important to note that the property is less likely to be found between three regions due the 7 edge requirement between regions. It would be interesting to find out what is the number of rigid clusters that can be merged in a sparse

network graph using merging algorithms, after which no significant increase can be found in size of the largest rigid region.

ACKNOWLEDGEMENTS

The author acknowledges Charles Welch for discussions on this topic.

REFERENCES

- [1] Chi-ChangChen, Chi-YuChang & Yan-NongL (2013) "Range-Free Localization Scheme in Wireless Sensor Networks Based on Bilateralation", International Journal of Distributed Sensor Networks , Article ID 620248, 10 pages
- [2] Deepak Prashar & Kiran Jyoti, (2019) "Distance Error Correction Based Hop Localization Algorithm for Wireless Sensor Network" Wireless Personal Communications 106, pp.1465–1488
- [3] T. Eren, O. Goldberg, W. Whiteley, Y. R. Yang, A. B. Morse, B. Anderson & P. Belhumeur, (2004) "Rigidity, computation, and randomization in network localization, in: Proceedings", IEEE INFOCOM Vol. 4, pp. 2673–2684.
- [4] T. Eren (2016), "Graph invariants for unique localizability in cooperative localization of wireless sensor networks: Rigidity index and redundancy index" Ad Hoc Networks, 44, pp 32-45
- [5] D. J. Jacobs & B. Hendrickson (1997) An algorithm for two-dimensional rigidity percolation: the pebble game, Journal of Computation Physics 137 (1997) 346–365.
- [6] X. Ji, H. Zha, (2004), "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling", I, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Vol 4, pp.2652 - 2661
- [7] Jyoti Kashniyal ,Shekhar Verma & Krishna Pratap Singh, (2019) "A new patch and stitch algorithm for localization in wireless sensor networks", ,Wireless Networks" Vol. 25, No 6, pp 3251–3264
- [8] Saroja Kanchi & Charles Welch, (2013) "An Efficient Algorithm for Finding Large Localizable Regions in Wireless Sensor Networks" Procedia Computer Science Volume 19, pp. 1081-1087
- [9] Saroja Kanchi & C. Wu, "Distributed Algorithm for Maximal Rigid Region in Sparse Wireless Sensor Networks", (2010) IEEE Asia-Pacific Services Computing, pp.400-404
- [10] G. Laman, "On graphs and rigidity of plane skeletal structures", (1970), Journal of Engineering Mathematics, vol 4, pp. 331–340.
- [11] Mort Naraghi-Pour & Gustavo Chacon Rojas, (2014) "A Novel Algorithm for Distributed Localization in Wireless Sensor Networks", ACM Transactions on Sensor Networks. Vol 11, pp.1-25.
- [12] Dragos Niculescu & Badri Nath, (2003) "Ad Hoc Positioning System (APS) Using AOA", Proceedings - IEEE INFOCOM, pp.1734 – 1743.
- [13] Slavisa, T, (2017). "Distributed algorithm for target positioning in wireless sensor networks using RSS and AoA measurements". Pervasive and Mobile Computing, 37, pp. 63-77.
- [14] Suman Pandey & Shirshu Varma (2016) "A Range Based Localization System in Multihop Wireless Sensor Networks: A Distributed Cooperative Approach" Wireless Personal Communications ,86, pp. 615–634

- [15] Soheil Salari, Il-Min Kim, & Francois Chan, (2018) ‘Distributed Cooperative Localization for Mobile Wireless Sensor Networks’, IEEE Wireless Communications Letters, vol. 7, no. 1,
- [16] Tan Wang , Hong Ding, Hui Xiong & Linhua Zheng, (2019), ‘‘A Compensated Multi-Anchors TOF-Based Localization Algorithm for Asynchronous Wireless Sensor Networks’’, IEEE Access, vol 7, pp.64162-64176
- [17] Tashnim, J. S. (2016). ‘‘Advances on positioning techniques for wireless sensor networks: A survey: ‘‘Computer Networks, 110 pp. 284-305.
- [18] Jing Wang, R. K. Ghosh & Sajal K. Das, (2010) ‘‘Rigidity, Computation, and Randomization in Network Localization’’, J Control Theory Appl , 8 (1) , pp 2–1.
- [19] Wu, Zhang, Cheng, S. Kanchi (2010), ‘‘Rigidity guided localisation for mobile robotic sensor networks’’, Journal of Ad Hoc and Ubiquitous Computing, vol 6, 2, pp.114-127.

AUTHORS

Dr. Kanchi is a Professor and the Interim Department Head of Computer Science at Kettering University. She works in the areas of Topological Graph Theory, Wireless Sensor Networks and Distributed Algorithms. She has published several articles in these areas.



METHODOLOGY TO EVALUATE WSN SIMULATORS: Focusing on Energy Consumption Awareness

Michel Bakni¹, Luis Manuel Moreno Chacón², Yudith Cardinale², Guillaume Terrasson¹, and Octavian Curea¹

¹Univ. Bordeaux, ESTIA Institute of Technology, F-64210 Bidart, France

²Universidad Simón Bolívar, Caracas, 1080-A, Venezuela

Abstract. *Nowadays, there exists a large number of available network simulators, that differ in their design, goals, and characteristics. Users who have to decide which simulator is the most appropriate for their particular requirements, are today lost, faced with a panoply of disparate and diverse simulators. Hence, it is obvious the need for establishing guidelines that support users in the tasks of selecting and customizing a simulator to suit their preferences and needs. In previous works, we proposed a generic and novel methodological approach to evaluate network simulators, considering a set of qualitative and quantitative criteria. However, it lacks criteria related to Wireless Sensor Networks (WSN). Thus, the aim of this work is three fold: (i) extend the previous proposed methodology to include the evaluation of WSN simulators, such as energy consumption modelling and scalability; (ii) elaborate a study of the state of the art of WSN simulators, with the intention of identifying the most used and cited in scientific articles; and (iii) demonstrate the suitability of our novel methodology by evaluating and comparing three of the most cited simulators. Our novel methodology provides researchers with an evaluation tool that can be used to describe and compare WSN simulators in order to select the most appropriate one for a given scenario.*

Keywords: *Methodology, Simulators, Wireless Sensors Networks, Energy Consumption.*

1 INTRODUCTION

In computer networks, network simulation is one of the most used and powerful evaluation methodologies. It has been used for the design and development of communication architectures and network protocols, as well as for verifying, managing, and predicting their behaviors. Since Wireless Sensor Networks (WSNs) provide a mean to capture and understand the reality and to interact on response of the gathered data [39], they have gained attraction in the research domains. Thus, simulators are also useful tools to evaluate WSNs.

In the last decades, several simulators have been either extended to include WSNs or built as WSNs simulators from the beginning. Different research groups develop different simulators according to their needs. For example, some simulators are designed to simulate the entire system. These simulators focus on the scalability, thus, their performance is a cornerstone in this regard. Others concern with the structure of the node and its energy consumption. For these simulators, wireless propagation and energy consumption modelling is what attracts attention.

Therefore, users who have to decide which simulator is the most appropriate for their particular requirements, are today lost, faced with this panoply of disparate

simulators. Thus, it is obvious the need of establishing guidelines and a systematic approach that support users in the tasks of selecting and customizing a simulator to suit their preferences and needs. To support this decision making, some works have tried to evaluate and compare several simulators and shyly propose some guidelines and steps to carry out such systematic evaluation [5][7][11][12][14][19][21][24][28][30][32][33][35][38]. However, as far as we know, there is not a generic methodological process to evaluate and compare network simulators that can be applied independently of their types and simulation scenarios.

In our previous works [2][3], a generic and novel methodological approach to evaluate network simulators is proposed. This approach is based on a set of qualitative and quantitative criteria. Even though, this methodological approach is generic and efficient to evaluate network simulators, it still lacks criteria related to WSN. Thus, the aim of our work, presented in this paper is threefold: (i) **extend the innovative and generic methodological approach** to include the evaluation of characteristics of WSN simulators, such as scalability and energy consumption awareness; (ii) **elaborate a study of the state of the art of WSN simulators**, with the intention of identifying the most used and cited in scientific articles; and (iii) **demonstrate the suitability of our innovative methodology** by evaluating and comparing several of the most cited WSN simulators.

The application of our methodological approach to evaluate three of the most cited simulators leads to results that are measurable and comparable. It allows a comprehensive overview of simulators features, advantages, and disadvantages. Therefore, this generic methodological approach provides researchers with an evaluation tool that can be used to describe and compare WSN simulators in order to select the most appropriate one for a given scenario.

2 RELATED WORK

The flexibility and validation in model construction offered by network simulation has fostered the research and development of multiple and different simulators. Thus, in order to select an appropriate network simulator, it is important to have good knowledge about their strengths and weaknesses, as well as to know how reliable are the models used by the simulators. In particular, for WSNs it is important to evaluate the scalability and energy consumption awareness of simulators.

To support this selection process, some works have proposed comparative criteria to carry out the evaluation of network simulators. For WSNs, the most recent and cited comparative studies are [5][7][11][12][14][19][21][24][28][30][32][33][35][38]. Most of them propose generic comparative qualitative criteria, such as type of simulator, API, languages supported, platforms supported, licenses, network support type, user interface [7][11][24][28][30][38]. Only the works proposed in [30][32] consider quantitative criteria, such as CPU utilization, memory usage, execution time, and scalability. Other studies also consider energy consumption modelling (e.g., wireless propagation, power consumption, battery, topology, antenna, radio

propagation, noise, and application modelling) and the challenges that face their implementations [5][12][14][21]. Few of such works are dedicated to evaluate WSN simulators in function of the energy consumption of each component of the WSN nodes and how they model the energy consumption of each component [19][35].

All these works, mainly evaluate WSN simulators based on a set of qualitative criteria, related and not related to scalability and energy consumption, but they do not establish any methodological process to perform the evaluation. We consider most of the same generic qualitative criteria, plus quantitative criteria, to evaluate any type of network simulators, as well as specific criteria to evaluate WSN simulators, such as scalability and energy consumption awareness. Besides, we propose a methodological approach to make such evaluation in a systematic and formal way.

The work proposed in [33], timidly proposes a methodology. However, the proposed guidelines and steps are focused on performing the network simulation, by following these steps: (i) evaluate the simulator based on a set of generic criteria (e.g., general features, visual support, flexibility, user support); (ii) select benchmarks to evaluate the simulated scenarios (e.g., network design, network protocols); (iii) conduct the simulation process; (iv) evaluate and analyze results. This methodology is focused on how to perform the simulation process; while our methodology, besides of considering such aspect, is intended to be generic, flexible, and suitable to support the selection of the most appropriate network simulator for a target simulation scenario, according to the user preferences and requirements.

3 WSN SIMULATORS: STATE-OF-THE-ART

We present a study of WSN simulators that are used in current researches.

3.1 Systematic Review

In order to find, select, and analyze the most popular and recent WSN simulators, we have followed a systematic review consisting of three main steps: (i) search of works dealing with WSN simulators; (ii) selection of relevant articles; and (iii) statistically find simulators cited in the set of the selected papers.

For the first step, the search was done on the Google Scholar search engine, which provides links to scientific repositories such as IEEE Xplore, ACM, and Springer. The search was based on tags that included the keys *WSN* and *simulator*, combined with tags related to the focus of the papers, such as *Survey*, *Review*, *Comparison*, *Evaluation*. We obtained more than 50 scientific articles.

For the second step, we select the most relevant articles related to WSN simulators evaluation, proposal, and comparison. From the more than 50 scientific papers obtained in the first step, some of them do not focus on simulators, but on designing and evaluating WSNs. We select works from 2010 and some older ones that have been widely cited. The final result was 37 relevant papers, categorized according to their main focus: (i) comparison papers, that evaluate and compare different

simulators; (ii) survey papers, in which authors present a general review of WSN simulators; (iii) simulator specific papers, which introduce the design or features of a particular WSN simulator; and (iv) trend papers, which contain explanation of the definitions and trends of how researchers evaluate WSN simulators.

In the third step, we analyze the selected papers and present statics of referenced WSN simulators on each category.

3.2 Categories of scientific articles

The selected papers were classified in four groups:

Comparison papers, which include comparative studies of WSN simulators, based on self-defined criteria that evaluate on each simulator, in order to analyze the differences among them [5][7][11][12][14][19] [21][24][28][30][32][33][35][38].

In [11], authors make a review of some of the open source network simulators (i.e., NS2, NS3, OMNeT++, and JSIM), comparing them according to languages supported, platforms supported, licenses, network support type, user interface, and API. In [30], authors compare NS2, NS3, OMNeT++, and GloMoSim. A unified scenario is applied by simulating a MANET routing protocol, in order to measure memory usage, computational time, and scalability, from which NS3 demonstrates the best performance. Similarly, in [7][21][28][32][33][38], some of popular WSN simulators (NS2, NS3, TOSSIM, OMNeT++, JSIM, Castalia, QualNet, EmStar, ATEMU, Avrora, SENS, COOJA, etc.) are described and compared based on the their general characteristics, their merits, and their limitations. The studies presented in [5][24], evaluate more than 20 simulators.

In [12], authors make a survey of available tools to evaluate WSN applications. They identify a set of models that are necessary to have in a WSN simulator: wireless propagation model, fine-grained energy expenditure model, non-linear battery model, and application model. In [14], authors compare Castalia, TOSSIM, and NS3 based on the sustainability to test dynamic network reconfiguration protocols. One of the topics that they evaluate is the energy consumption model of the simulators. They identify that the ability to model the RF states of the sensors is important to model the energy of sensors. In [35], authors compare Castalia, MiXiM, TOSSIM, and WSNnet, based on topology, antenna, radio propagation, noise, RF, medium access control, and energy consumption modelling. They execute a series of real experiments and calibrate the radio propagation model and the energy consumption model. In [19], authors compare NS2, NS3, TOSSIM, and OMNeT++, focusing on the modelling of the energy consumption. They describe the energy consumption of each component of the WSN nodes and show how the studied simulators model the energy consumption of each component.

Survey papers, that describe WSN simulators in a general way, but there is no comparison among them [1][9][10][15][16][22][36][40]. In [29], a review of network modelling and simulation tools is presented, including WSN simulators, such as

NS2, OPNET, and GloMoSim. Authors in [17] present a review of several WSN simulation tools. They mostly focus on their suitability for large-scale WSNs.

Simulator-specific papers, which focus on describing new WSN simulators[13][18][20][23]. In [23], the support for heterogeneous networks in IDEA1, is presented. In [13], WebShawn, an online WSN simulator is presented. A4WSN, an architecture-driven modelling platform for analysing and developing WSNs is presented in [18]. NS2 is the simulator used in [18] to analyse S-MAC and leach in WSNs.

Trends papers focused on studying proposed approaches to evaluate WSN simulators and research trends. In [26], authors compile a large set of papers of wireless communication-related conferences and review the statistics about the tools (i.e., testbeds and simulators) the researchers use to evaluate their experiments. Additionally, they address the issues and challenges facing the proper use of WSN simulators. They assert that simulators do not reproduce actual environmental conditions of deployed systems, thus experimental testbeds can be developed to replace simulators. In [6], authors discuss topics to consider when addressing IoT issues. They present the research trends on IoT simulators in the last years. To achieve that, authors describe existing tools that are used by researchers to prove and evaluate their findings on IoT research. They claim that more work is needed to conduct large-scale, robust and effective IoT simulation, and prototype evaluations.

3.3 Statistical Analysis of Selected Papers

In total, in the selected papers there are 369 citations, distributed among more than 100 WSN simulators. According to the number of citations, simulators are categorized into three groups: (i) Group 1, composed by simulators with more than 12 citations; this group includes 7 simulators; (ii) Group 2, involves all simulators with 6 to 12 citations; it contains 12 simulators; and (iii) Group 3, covers all simulators that are cited less than 6 times; it contains 87 simulators. Figure 1 presents the number of citations and the number of simulators of each group.

The total sum of citations that have the simulators of the Group 1 is 125, which represent 33.88% of the citations distributed in 7 simulators. Group 2 has in total 117 citations, which means the 31.71% of the citations. Group 3 has 127 citations, which represents a 34.42% of the total of citations. Figure 1 shows that Groups 1 and 3 contain more citations than Group 2. In the case of the Group 3, those citations are distributed in a larger number of simulators.

Figure 2 shows the number of citations of the simulators of Group 1, in which the most cited simulators are NS2, TOSSIM, and OMNeT++. NS2 and TOSSIM are presented in 24 papers, and OMNeT++ is presented in 20 papers.

This study can help to identify the most used WSN simulators, but the number of citation is not enough to provide comparison-based view. Therefore, a more robust approach to compare and evaluate WSN simulators is needed. In next sections, our methodology is presented. Then, in order to show its applicability and suitability, the most cited simulators are evaluated and compared.

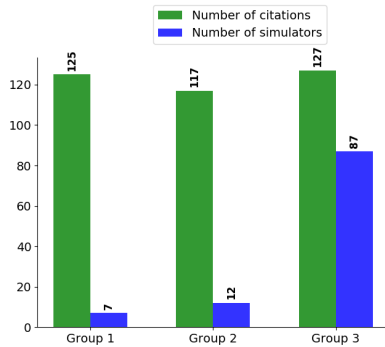


Fig. 1. Citations of WSN simulators

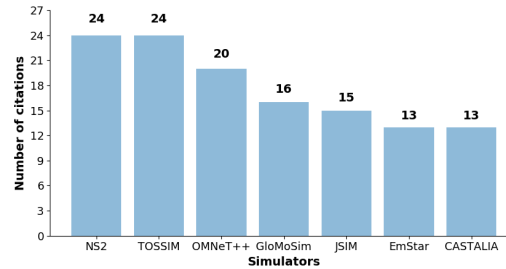


Fig. 2. Most cited simulators.

4 METHODOLOGY TO EVALUATE WSN SIMULATORS: OUR PROPOSAL

The methodology proposed in our previous works [2][3] does not include qualitative criteria to analyze the simulator scalability and the support of simulators on evaluating traces of energy consumption, sensor nodes mobility, and wireless medium modelling. These characteristics are present in WSN and are less important for general networks. Therefore, considering features that characterize WSN simulators, we propose an extension of the previously proposed methodology.

4.1 Methodological process

The methodological process consists on the following steps:

Step 1. Establish a set of criteria. The evaluation of the simulator requires clear and accurate criteria to assess the different aspects of the simulator. Qualitative criteria can be described by a word or number, while quantitative criteria need to be measured.

Step 2. Establish the experimental setup. The platform in which simulators are installed to be evaluated should not be neglected. Using a specific simulator, the way the operating systems manage system resources and the produced overhead have an important impact on the performance and behavior of such simulator. Hence, it is worthy to install the simulators on different systems (e.g., Windows, Linux, MacOS) under the same architecture.

Step 3. Evaluate the qualitative criteria of the simulator(s). To comply this step, it is recommended to revise the available documentation of simulator(s) and elaborate a table highlighting their characteristics.

Step 4. Design a test scenario to evaluate the measurable criteria. In a data network, a scenario is defined by a set of parameters that characterize a specific use case. According to the protocols that are intended to evaluate, it is important to decide the network elements to be simulated, the number and the type of experiments, as well as the time of the simulation, taking into account the

criteria to be evaluated.

Step 5. Evaluate the measurable criteria of the simulator(s) by executing the designed experiments. In order to obtain the results, each designed scenario has to be implemented on the simulator(s). In order to facilitate the analysis and comparison (if it is the case), results must be presented in tables and graphics.

Step 6. Elaborate a discussion by analyzing the results.

With this six-steps methodology, users can evaluate network simulators to select the most appropriated according to their needs and scenarios. For the comparative analysis, we also propose a set of criteria, which complement the Step 1.

4.2 Criteria used in the methodology

The set of criteria used in the methodology are as follows:

1. Nature of the simulator: The nature of the simulator is an assessment of how the simulation is performed. Precisely, the term *simulation* means that the simulated process is programmed and only the software aspect is involved in the simulation. But if the word *emulation* is used, the hardware is also involved in the simulation process [31].

2. Type of simulator: Network simulators are based on two philosophies, discrete-event simulator or trace-driven one. In the first case, an initial set of events is generated, representing the initial conditions. Those conditions, in turn, generate another set of events and so on. The process continues until the end of the simulation. This philosophy is highly used in WSNs, since it allows to easily simulate hundreds of jobs running on different WSN nodes [40]. The trace-driven simulation mostly plays a crucial role in real time applications. It provides information which lets users to have more detailed knowledge of the simulation model [24].

3. License: From a legal aspect, simulators can be private property or they can be used under a free or public agreement.

4. User interface: It is an evaluation of how a user can interact with the simulator. This criterion includes two aspects: (i) **Graphical User Interface (GUI):** Is it an integral part of the simulator? What is the level of details it can show?; and (ii) **Supported programming languages:** Can users interact with the simulator by programming scripts? Can users develop a piece of software to communicate with the simulator?

5. Supported platforms: It is the characterization of the usability of the simulator source code on different platforms and operating systems.

6. Heterogeneity: It is an evaluation of the ability to simulate heterogeneous systems where different types of nodes can exist in the same scenario.

7. Level of details: It consists on evaluating the level of aspects that are being simulated. Sorted in descending order, they are: abstract algorithms, high level protocols, low-level protocols, and hardware. The lower the level is, the less the

assumptions is and the more the constraints are.

8. Modelling: It represents the ability to modify existing models in the simulator or to implement and test new ones.

9. Mobility modelling: It indicates the support of the WSN simulators for modelling mobility of sensor nodes.

10. Wireless medium model: It means the ability to model the wireless medium used to estimate the radio signal strength, quality, and delay between transmitter and receiver units.

11. Energy consumption modelling: It refers to the current draw at the sensor node level. A node sensor is typically composed by four major components: sensing unit, the power supply unit, the processing unit, and the communication unit. Units that consume energy are the sensing, the processing, and the communication units; while the power supply unit, which can embed energy harvesting solutions, provides energy. The aspects considered to evaluate this criterion are: (i) **Battery modelling:** which consists on evaluating if non-linear or linear battery models are implemented in the simulator; (ii) **RF states modelling:** that represents the capacity to consider and simulate all RF states, such as Idle, Sleep, Receiving, and Transmitting; and (iii) **Limitations:** to address the constraints of simulators regarding power consumption modelling. With these parameters, a scenario can be designed to evaluate energy consumption, such as total power consumption or the energy consumption on each RF state.

12. Supported technology and protocols: To evaluate the support provided for the protocols, TCP/IP model is used [34].

13. Measurable criteria: The main purpose of the measurable criteria is to provide a general idea of the effectiveness of the simulator in terms of the consumption of available resources, scalability, and energy consumption modelling capacities. Our methodological approach includes four factors for the simulator performance study, for both network and WSN simulators: (i) **CPU Utilization:** it is a measure of the simulator performance [4], which consists in the percentage of time spent performing the simulation process of the total processing time [25]; (ii) **Execution time:** it is the time needed to complete a simulated scenario; measured in seconds; (iii) **Memory usage:** it is the amount of memory used by the simulator, measured in bytes; and (iv) **Scalability:** it is the measure of the capacity of the simulator of simulating huge scenarios; how many network or WSN components can be simultaneously simulated without degrading the simulator performance?

All these measurable aspects evaluate the simulator(s) performance. However, for WSN simulators there are other measurable aspects that evaluate their capacity of modelling energy consumption of WSN nodes. Hence, this measurable criterion is considered only for WSN simulators: **Trace of nodes energy consumption:** it represents the obtained measures of energy consumption of WSN components, ac-

ording to the modelling allowed by the evaluated simulator(s). This criterion does not measure the performance of the simulator itself, but its capacity of measuring the energy consumption of WSN nodes.

In [3], we apply the methodology to evaluate Packet Tracer simulator. In [2], the goal was to compare data network simulators on a standard base, thus, the selection of one simulator over the others can be justified. The approach was applied on GNS3, a data network emulator, as well as on Packet Tracer. In this work, in order to show how to apply this extended methodological approach for WSN simulators, we evaluate and compare three of the most cited ones.

5 APPLICATION OF THE PROPOSED METHODOLOGY

To show the suitability and flexibility of the extended methodology, we apply it to evaluate and compare the most cited WSN simulators, identified in Section 3: NS2, TOSSIM, and OMNeT++. NS2 is a discrete event network simulator, initially designed to simulate wireless LAN protocols, though later was expanded to mobile ad-hoc networks support. TOSSIM, stands for Tiny OS simulator, is a discrete-event simulator for TinyOS applications. TOSSIM is a TinyOS emulator, but not a general WSN simulator [1]. OMNeT++ (for Objective Modular Network Testbed) is a discrete event simulator based on C++. In this work, OMNeT++ is used with INET, which is an OMNeT++ framework that has implemented models for wired, wireless, and mobile networks. In the following, we explain how the methodological process was applied to evaluate these three WSN simulators.

Step 1: Establish a set of criteria. As it is illustrated in Table 1, the set of criteria considered are the ones described in Section 4.

Step 2: Establish the experiment setup. To evaluate the considered simulators in different systems, they are tested on Linux Ubuntu 16.04 LTS and Microsoft Windows 10 version 10.0.14393. They were installed on the same computer with the following characteristics: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz with 16 GB of RAM, 915 GB of disk allocated for Linux, while 909 GB is allocated for Windows.

Step 3: Evaluate the qualitative criteria. The qualitative criteria of the three simulators are summarized in Table 1. We particularly comment about energy consumption modelling of each simulator. TOSSIM does not have a model of the energy consumption. However, PowerTOSSIM z [27], an extension of TOSSIM, adds that functionality. The energy consumption model of PowerTOSSIM z considers four hardware components that consume energy: microprocessor, LED, RF module, and memory. Additionally, PowerTOSSIM z models modern batteries, by simulating their nonlinear behavior, expressed by the effects of rate capacity and recovery capacity [14]. PowerTOSSIM z has disadvantages as well: (i) it is a plug-in to TOSSIM, i.e., PowerTOSSIM z cannot dynamically change the fixed power-consumption parameters during the simulation [14]; (ii) the energy consumption caused by power-state transitions and the time needed for that, are not taken into

account; (iii) PowerTOSSIM z cannot simulate energy harvester units; and (iv) this plugin has a poor level of support. The project is not longer maintained.

The energy consumption model in NS2 maintains a total value for the energy stocked at each node in the WSN. When the energy of a node is completely consumed, the model declares it as *dead*. The model considers only the consumption of the RF module, it is based on a state machine that has the following states for the RF model: Idle, Sleep, Receiving, and Transmitting. NS2 cannot simulate non-linear batteries. The batteries of the sensor nodes are ideal batteries. Moreover, NS2 energy model does not support energy harvesting or battery recharging. It monitors the changes in the power level in one way only, i.e., the consumption.

The energy modelling in INET is divided in 3 parts: energy consumption models, energy generation models, and energy storage models. These models can represent the energy in two different ways: charge and current (CC) or energy and power (EP). Energy consumer models describe the energy consumption of units and node components over time. There are three models of energy consumption. Two of them are based on the RF states and they differ in the way the energy is represented (EP or CC). The third model is a basic model that represents the energy consumption of a node with two states only (Idle and Active). It is used to have a general overview of the energy consumption without focusing on the details.

Regarding the mobility modelling, NS2 and OMNeT++/INET support simulations for mobile nodes. TOSSIM does not have this feature. However, an extended plug-in for TOSSIM can be added to provide the support for the mobility, it is called MOB-TOSSIM [8].

Step 4: Design the test scenarios. Basic scenarios are designed to evaluate the performance of the selected simulators and their energy consumption modelling capacity. The performance is measured in terms of CPU utilization, memory usage, execution time, and scalability. A meshed topology is adopted for the WSN, whose size is increasing exponentially for different tests. The basic component (BC) of the topology consists of four sensor nodes, each one placed in the vertex of a 10 meters x 10 meters square. The first test includes only one BC with four sensor nodes. The second test is done with two BC, i.e., eight nodes. The third one is composed by four BC, with 16 nodes, and so on. In total, eight simulations take place on each system (Linux and Windows), with the number of BCs changing as: 1, 2, 4, 8, 16, 32, 64, and 128 for each simulator.

Each node in the WSN is configured to use IPv4 and ICMPv4. The goal is to create a data message with an echo request to all other nodes in the topology. A node that receives the echo request, replies back the same message. Each simulation lasts 100 seconds. The frequency is 1 Hz, which means that one echo message is sent every second. As a result, there are 100 echo request messages sent per simulation.

To evaluate the energy consumption models, another test scenario is proposed. This scenario consists of two nodes, which are 10 meters apart from each other.

Table 1. Comparison of WSN simulators using the proposed criteria

Criterion	TOSSIM	NS2	OMNeT++/INET
Nature of the simulator	Emulator	Simulator	Simulator
Type of the simulator	discrete-event	discrete-event	discrete-event
License	BSD-license	GNU GPLv2 license	Academic Public License. INET models are licensed under LGPL or GPL.
User interface	GUI: Yes, through TinyViz. Supported programming language: Python, C++ and NesC	GUI: Yes, through Nam. Supported programming language: C++ and OTcl	GUI: Yes, a built-in GUI interface is available Supported programming language: C++ and NED
Supported platforms	Linux and Windows	Linux, MacOS and FreeBSD	Windows, Linux and Mac OSX
Heterogeneity	No	No	Yes
Level of details	Code level	Packet Level	Packet level
Modelling	Available	Available	Available
Mobility model	Yes, through MOB-TOSSIM	Yes	Yes
Wireless medium model	Path loss models: lognormal shadowing Other models: noise modelling	Path loss models: shadowing, ray ground, free space	Path loss models: free-space, log-normal shadowing, rayleigh fading, 2-ray ground, rician fading, nakagami fading Other models: Background noise, obstacle loss and propagation models
Energy model	Battery model: No RF states: Yes Limitations: Cannot model energy harvester units	Battery model: Only for Ideal Battery RF states: Yes Limitations: Cannot model sensing and processing units	Battery model: Yes RF states: Yes Limitations: Cannot model sensing and processing units
Supported technology and protocols	TOSSIM simulates entire TinyOS applications, including the network stack that supports TinyOS implementation.	Application Layer: DHCP, telnet, FTP, HTTP Transport Layer: TCP, UDP, SCTP, XCP, TFRC, RAP, RTPM Network Layer: IPv4, IPv6 Link Layer: HDLC, GAF, MPLS, LDP, Diffserv, DropTail, RED, RIO, IGMPv2, IGMPv3, WFQ, SRR, Semantic Packet Queue, IPv6, MCoA, MIPv6, REM, CSMA, 802.11b, 802.15.4, Satellite Aloha Routing Protocols: RIP, AODV, Click, DSDV, DSR, NixVectorRouting, OLSR	Application Layer: HTTP, DHCP, BitTorrent, P2P, Video Streaming, Voice Transport Layer: TCP, UDP, SCTP, RTP, RTCP. Network Layer: ARP, HIP, IPv4, IPv6 Link Layer: 802.11, 802.11p, 802.1e, WiMAX, 802.11 Routing Protocols: AODV, BGP, GPSR, link-state routing, OSPF, OSPFv2, PIM, RIP

Table 2. Parameters of the energy consumption scenario

Parameter	802.11b	802.15.4
Bitrate	11 Mbps	250 Kbps
MAC	CSMA/CA with RTS/CTS CSMA/CA with CCA	
Transmitting power	750 [mW]	52 [mW]
Receiving power	220 [mW]	59 [mW]
Sleep power	0.2 [mW]	0.06 [mW]
Idle power	0.2 [mW]	0.06 [mW]

One of the nodes is periodically sending an ICMPv4 echo request to the other node. When the other node receives the request, it replies back the same message. The echo request and the echo reply are identical in length and format. Therefore, the energy consumption of both nodes will be the same. The communication of nodes is made using two different wireless link protocols: 802.11b and 802.15.4. For each protocol the payload length of the ping message starts at 10 bytes, then, it is gradually being increased by 10 bytes, until the payload size reaches 90 bytes. In total, there are 9 simulation per protocol. Each simulation is repeated 3 times for different values of frequency of the ping messages: 0.1, 1, and 2 Hz.

For the 802.11b scenarios, the energy consumption parameters were taken from the data sheet of HDG204 RF Module¹, while for the 802.15.4 scenario was used the data sheet of CC2420 RF Module². Each simulator was configured to use the models of the protocols with the values of the standards. The values of the energy consumption for each module is shown in Table 2.

Step 5: Evaluate the measurable criteria. NS2 is only evaluated in Linux, since it is the only platform that supports its installation. OMNeT++/INET is installed on both Windows and Linux. The NS2 version used is the 2.35³, for OMNeT++, it is 5.4.1⁴, and for INET, it is 4.1.0-810053f713⁵. TOSSIM (PowerTOSSIM z) is not installed in none of the systems, since it has a poor level of support for the recent OS versions and it is not possible to install the simulator on the systems used. Thus, PowerTOSSIM z is not evaluated in terms of measurable criteria.

Performance scenarios: In the performance scenario, the CPU utilization is evaluated for the simulators during 100 seconds of simulation. The results of the performance evaluation of the CPU utilization for different BCs are shown in Figure 3. NS2 tends to consume all available CPU cycles, whatever the number of the BCs is, while OMnet++ consumes the CPU differently in Linux than in Windows. Figure 3 shows that the CPU utilization in Windows is always less than Linux when the same scenario is implemented. In both OSs, as the number of BCs increases, the average value of CPU utilization increases as well.

¹ <https://media.digikey.com/pdf/DataSheets/H&DWireless0PDFs/HDG204DS.pdf>

² <http://www.ti.com/lit/ds/swrs041c/swrs041c.pdf>

³ <https://sourceforge.net/projects/nsnam/>

⁴ <https://github.com/omnetpp/omnetpp/tree/omnetpp-5.4.1>

⁵ <https://github.com/inet-framework/inet/tree/v4.1.0>

Figure 4 represents the results of memory usage for both simulators on a logarithmic scale as the number of BCs increases. NS2 shows proper memory usage when the BCs are 4 or less. After that, the usage tends to follow an exponential orientation. On both operating systems, OMNets++ shows a strictly controlled memory usage as the number of the BCs increases. The memory usage in Windows shows lower values compared to Linux when the same scenario is being implemented.

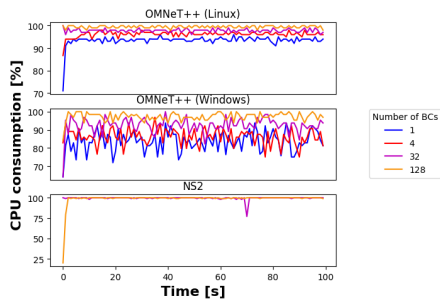


Fig. 3. CPU utilization of NS2 and OMNeT++

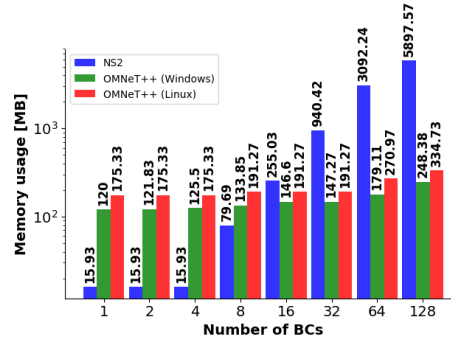


Fig. 4. Memory usage of NS2 and OMNeT++

In order to obtain the execution time in OMNeT++, the express-mode is used, since the normal mode was intentionally built to run slowly to allow the user to trace the events that are occurring during the simulation. Figure 5 represents the execution time for the simulators on a logarithmic scale. We note that NS2 has lower execution time for the scenarios with less than 16 BCs, while OMNeT++ has lower execution time for the scenarios that have 16 BCs or more. The execution times of OMNeT++ in Windows and Linux are similar.

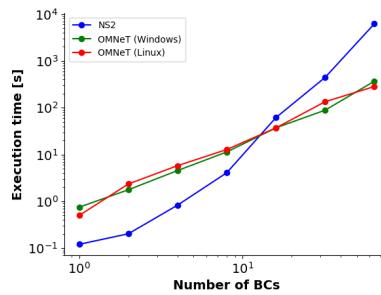


Fig. 5. Execution time of NS2 and OMNeT++

Scalability, as the capacity of supporting scenarios with a huge quantity of WSN components, can be deduced from the CPU utilization, memory usage, and total execution time in terms of number of BCs. Results shown on Figures 3, 4,

and 5 demonstrate that OMNeT++ scales better than NS2. Even though the CPU utilization of OMNeT++ increases as the number of BCs increases, it is comparable to the CPU utilization of NS2 for the largest scenario (Figure 3), its memory usage increases less than NS2 for larger scenarios (Figure 4), and its total execution time is linear in contrast to the super-linear execution time of NS2 (Figure 5).

Energy consumption scenarios: The main objective of the energy consumption scenario is to demonstrate the information that can be obtained from the two simulators. To do so, the same scenarios were implemented on them.

NS2 has only a command-line interface; thus the output is text displayed on the terminal. Information related to energy consumption is not included. Thus, we developed an animator that was integrated to NS2 as a plug-in, in order to control the simulation time, to capture the output, and to extract the energy consumption information. On the other hand, OMNeT++ stores information about the simulations in files, that can be exported in multiple formats for later data processing. OMNeT++ shows the same results both on Windows and Linux, regarding the energy consumption evaluation. Therefore, the results of the energy consumption scenario in OMNeT++ are presented only once and without mentioning the OS.

The energy model in both simulators trace only the energy consumption of RF module, i.e., the consumption of the node CPU and the sensors are not included.

By comparing the results for the same scenarios obtained from NS2 and OMNeT++, there are differences and similarities. In the 802.11b scenarios, both simulators have the capability to accurately simulate the CSMA/CA mechanism, including parameters of PHY and MAC layers of each frame sent during each phase of the mechanism, such as RTS and CTS frames. Additionally, the data and acknowledgement (ACK) frames are simulated as well. The implementation uses the standard guideline to define the length of each frame used in the protocol, as well as the preamble length and the PHY header. The time spent sending RTS, CTS, and ACK frames are similar for both simulators as shown in Table 3. But the time spent to send data frames is higher in the OMNeT++ simulator as Table 4 shows.

Table 3. Time spent in 802.11b for control frames

Frame type	Time in OMNeT++ [μs]	Time in NS2 [μs]
RTS	207	207
CTS	203	202
ACK	203	202

Table 4. Time spent in 802.11b for data frames

Payload [$Byte$]	Time in OMNeT++ [μs]	Time in NS2 [μs]
10	246	239
30	261	253
50	275	268
70	290	282
90	304	297

By examining one of the repeated interval of the simulation (i.e., the time that includes sending one ping messages), the results show that the energy consumption

of both simulators are not the same. Figure 6 shows the energy consumption in an interval when the frequency is 1 Hz, for the 802.11b scenario, for both simulators. Each pair of columns represents a payload size; the columns to the right is for results obtained from OMNeT++, while the column to the left is for results obtained from NS2. In general, when the same scenario is implemented, the reported energy consumption in OMNeT++ is slightly higher than NS2.

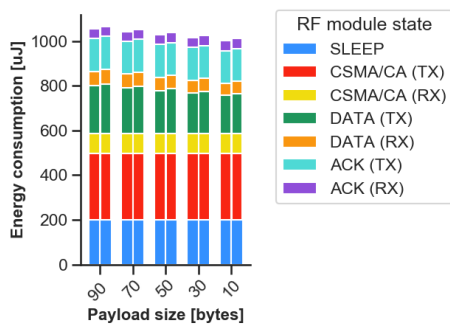


Fig. 6. Energy consumption results of simulations using 802.11b for NS2 and OMNeT++

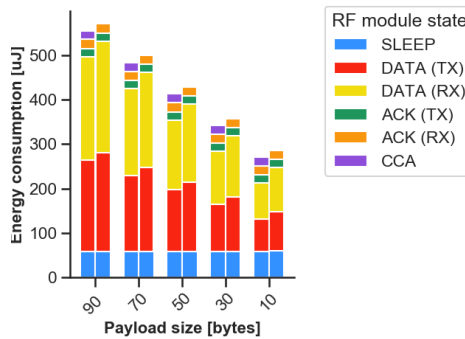


Fig. 7. Energy consumption results of simulations using 802.15.4 for NS2 and OMNeT++

In the 802.15.4 scenarios, both simulators present different implementations of the protocol, this, in turn, affects the energy consumption. Figure 7 shows the energy consumption for one interval of 802.15.4, when frequency is 1 Hz.

In both simulators, an access method is implemented for 802.15.4, it is called Clear Channel Assign (CCA). However, this mechanism is not linked to the energy model in OMNeT++, i.e., using this mechanism does not consume energy. On the other hand, in NS2, CCA is linked to the energy model. As a result, in term of energy consumption, in NS2 there is one more state for the RF model, that is shown in Figure 7 as a violet rectangle called CCA.

Step 6: Elaborate a discussion. From the methodological process, it is possible to detect advantages and disadvantages of the three analyzed WSN simulators. NS2 is a generic data network simulator that was later adapted to suit WSN, while OMNeT++, was built to support the WSN from the beginning. TOSSIM, is an emulator for TinyOS, which is an OS widely used for embedded systems.

The principal drawback of TOSSIM is that it is not compatible with the modern systems. It was not possible to install and run the evaluation scenarios in TOSSIM. Therefore, only the qualitative parameters are available for the comparison with the other simulators. Although, NS2 is only supported on Linux, and despite the fact that it is no longer maintained in favor of NS3, it is one of the most cited simulators in the research domain. This work shows that NS3 has not completely

replaced NS2. NS3 is still in development and many protocols supported in NS2 have not been yet implemented in NS3. OMNeT++ is supported both on Linux and Windows. The project is still maintained and the simulator is regularly updated.

From the performance point of view, the obtained results showed that NS2 is more suitable for projects with less than 128 nodes. In these cases, the execution time in NS2 takes place for a short period of time. When the number of nodes is equal to or greater than 128, the simulation scenario lasts in NS2 longer than in OMNeT++. Moreover, the memory used for scenarios in NS2 with more than 128 nodes are much larger than the memory used in the OMNeT++ simulations for the same scenarios. OMNeT++ is more stable in terms of scalability. As the number of nodes increase, the memory usage increase. However, the CPU utilization is similar for all simulation scenarios. Moreover, the execution time in OMNeT++ using the fast-mode is greater than the execution time in NS2 for the scenarios with less than 16 BCs and it is shorter in the scenarios with 16 or more BCs, which evidence its better ability to scale. By analyzing these results, we note that there is a clear pattern for CPU utilization. Whatever the OS is and whatever the complexity of the scenario. Even though OMNeT++ consumes CPU less than NS2 when the same scenario is implemented, the simulator tends to consume almost all available CPU cycles. On the other hand, as the complexity of the scenarios is increasing, the memory usage increases slightly. Although the results of simulation in OMNeT++ are identical when the same scenario is implemented in Windows and Linux, performance results in Windows show a better CPU and memory utilization.

Both simulators are evaluated using scenarios that are developed to verify the energy consumption models. Results show that they use a similar philosophy to model the energy consumption in a WSN node. The concept is totally based on the RF states of the transceiver, i.e., NS2 and OMNeT++ consider only the energy consumption of the RF module. They do not take into account the consumption of other hardware devices, such as the node CPU or the node on-board sensors.

NS2 and OMNeT++ implement 802.11b with a high level of details. Although the implementations are close, there are still differences in the time spent sending and receiving the data frames. The same problem occurs in the scenarios of 802.15.4. As a result, for the two protocols, when the same scenario is implemented, the time spent sending and receiving data frames in OMNeT++ is always greater than that of NS2. Therefore, when the same scenario is implemented, the energy consumed in OMNeT++ is greater than the energy consumed in NS2.

Results of the energy consumption scenarios indicate that OMNeT++ has consistent results across the two platforms used in the evaluation, i.e., Windows and Linux. This is an important characteristic for simulators that operate across multiple platforms. NS2 is only supported on Linux platforms.

OMNeT++ has a built-in GUI. It allows the user to graphically run the scenarios and to easily debug the source code. It also has a built-in Integrated Develop-

ment Environment (IDE) that helps the developer to identify errors and to check the syntax before compiling the code. These features consume a large amount of resources on the host machine in which the simulator is running, which can be a drawback when working on hosts with limited resource.

6 DISCUSSION ABOUT THE PROPOSED METHODOLOGY

Nowadays, with the huge variety of available simulators, it is important to identify which simulator suits the most for a given scenario. The problem of selection always arises, no matter if the simulator is going to be used for academic purposes or industrial development. From the previous proposals [3] [2], we add criteria to address the evaluation of WSN simulators, in terms of their scalability and capability of modelling mobility, wireless medium, and energy consumption.

In a simulation environment, scalability is a subject governed by the hardware of the simulator host (CPU and memory). Hence, it is not addressed as a separated criterion. Instead, it is handled as a scenario parameter to observe how the number of the nodes can impact the performance in term of CPU utilization and memory usage. Thus, an approximate threshold for the number of nodes which makes decline the simulator performance can be detected. On the other hand, energy issues can be addressed using modelling technologies. Our proposed methodology addresses these issues: it proposes guidelines and criteria to measure the scalability of simulators and to evaluate their energy consumption awareness modelling.

Most WSN simulators models the energy consumption of the RF module. Although the RF activities are responsible for the major part of the energy consumption in the node, the consumption of CPU and sensors cannot be neglected. In [37], authors calculate the power consumption average of the sensor unit, the RF module, and the microcontroller for a WSN application. In their specific application the average of power consumed for the RF activities were 62%, the average of power consumed for the sensor and the microcontroller were 14% and 24% respectively; which means that the RF activities can consume more than the sum of the other units. Therefore, it is important for a simulator to model the energy consumption of all units present in the node in order to get an accurate estimation of the energy consumed by the node. Thus, our methodology evaluate all these aspects.

The proposed methodological approach is flexible, allowing to integrate another items to cover new aspects needed by users. For instance, it is possible to add criteria to evaluate the simulators capacity of modelling the antenna or the battery behaviour. By following the methodology steps, the advantages and disadvantages of one or more simulators for a certain application can be identified. Thus, the selection of one of them can be well justified and probed, as well as its suitability for specific user needs and scenarios.

Although the methodology provides a comprehensive method to compare WSN simulators, there are still aspects to be covered. For example, the study of energy modelling can be extended to include the support for the battery model. When

considering the estimation of the node lifetime, the model that trace the remaining energy is different from the one that trace the consumed energy. The support of parallel processing is another item that can be extended as well. This feature exists in some simulators and has a huge effect on performance.

Besides, wireless link protocols have special role in WSN. Thus, it is recommended to separate it from the protocol items and consider additional aspects that concerns the users of the simulators, such as the support of different bit rates and fragmentation. Finally, WSNs are still in developing and new technologies will be adapted. Thus, new features will be added and WSN simulators have to answer to that. Our methodology faces all these challenges by being extensible, flexible, and generic, and still being a powerful tool to evaluate and compare network simulators.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have address the difficulty of selecting a WSN simulator to fit a given scenario. To achieve that, we extend our previous proposed methodology, by integrating new criteria to address WSN evaluation, such as scalability and the modelling of mobility, wireless medium, and energy consumption.

In order to demonstrate the efficiency and suitability of our methodology, we elaborate the state of the art of WSN simulators, following a systematic review of most cited and recent scientific papers. From this review, we select the three most cited WSN simulators (i.e., NS2, TOSSIM, and OMNeT++) to evaluate and compare them following our proposed methodological approach. The application of the methodology proves that it does not only highlight general aspects of the simulators behaviors but it shows their disadvantages as well.

In a future study, we plan to include other evaluation criteria, such as the capacity of simulators for parallel processing and support of different bit rates and fragmentation. We are also working on proposing an energy consumption model to include the support for the battery behaviour modelling.

ACKNOWLEDGEMENTS

The work presented in this paper has been financially supported in part by the Regional Council of New Aquitaine (as part of the Call for Projects 2016 funds), in the frame of the OUDINI research project.

References

1. Abuarqoub, A., Hammoudeh, M., Alfayez, F., Aldabbas, O.: A Survey on Wireless Sensor Networks Simulation Tools and Testbeds, vol. 3, chap. 14, pp. 283–302. IFSA (01 2016)
2. Bakni, M., Cardinale, Y., Moreno, L.: Experiences on evaluating network simulators: A methodological approach. *Journal of Communications (JCM)* pp. 1–11 (2019)
3. Bakni, M., Cardinale, Y., Moreno, L.M.: An Approach to Evaluate Network Simulators: An Experience with Packet Tracer. *Revista Venezolana de Computación* **5**, 29 – 36 (Jun 2018)
4. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. *Computer* **40**(12) (2007)
5. Chéour, R., Jmal, M.W., Kanoun, O., Abid, M.: Evaluation of simulator tools and power-aware scheduling model for wireless sensor networks. *IET Computers & Digital Techniques* **11**(5), 173–182 (2017)

6. Chernyshev, M., Baig, Z.A., Bello, O., Zeadally, S.: Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal* **5**, 1637–1647 (6 2018)
7. Chhimwal, P., Rai, D.S., Rawat, D.: Comparison between different wireless sensor simulation tools. *IOSR Journal of Electronics and Communication Engineering* **5**(2), 54–60 (2013)
8. Derhab, A., Ounini, F., Remli, B.: Mob-tossim: An extension framework for tossim simulator to support mobility in wireless sensor and actuator networks. In: *Internat. Conf. on Distributed Computing in Sensor Systems*. pp. 300–305 (2012)
9. Du, W., Navarro, D., Mieleveville, F., Gaffiot, F.: Towards a taxonomy of simulation tools for wireless sensor networks. In: *Inter. Conf. on Simulation Tools and Techniques*. pp. 52:1–7 (2010)
10. Fahmy, H.M.A.: Simulators and emulators for wsns. In: *Wireless sensor networks*, pp. 381–491. Springer (2016)
11. G Gupta, S., Ghonge, M., D P M Thakare, P., Jawandhiya, P.: Open-source network simulation tools: An overview. *Internat. Journal of Advanced Research in Computer Engineering and Tech.* **2** (2013)
12. Garg, K., Frster, A., Puccinelli, D., Giordano, S.: Towards realistic and credible wireless sensor network evaluation. vol. 89 (09 2011). https://doi.org/10.1007/978-3-642-29096-1_4
13. Godoy, D., Sosa, E., Daz Redondo, R., Bareiro, H.: Webshawn, simulating wireless sensors networks from the web. pp. 190–195 (10 2017). <https://doi.org/10.1109/WiMOB.2017.8115829>
14. Helkey, J., Holder, L., Shirazi, B.: Comparison of simulators for assessing the ability to sustain wireless sensor networks using dynamic network reconfiguration. *Sustainable Computing: Informatics and Systems* **9**, 1–7 (2016)
15. Imran, M., Abas, S., Halabi, H.: A survey of simulation in sensor networks. *Information Technology Internat. Symp, IEEE* **2** (2010)
16. Kellner, A., Behrends, K., Hogrefe, D.: Simulation environments for wireless sensor networks. Tech. rep., Inst. of Computer Science – Georg-August-Universit at Göttingen (2010)
17. Khan, M.Z., Askwith, B., Bouhaf, F., Asim, M.: Limitations of simulation tools for large-scale wireless sensor networks. In: *Internat Conf on Advanced Informat Networking and Apps*. pp. 820–825 (2011)
18. Krishna, K.H., Kumar, T., Babu, Y.S.: Energy effectiveness practices in wsn over simulation and analysis of s-mac and leach using the network simulator ns2. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. pp. 914–920. IEEE (2017)
19. Lahmar, K., Cheour, R., Abid, M.: Wireless sensor networks: Trends, power consumption and simulators. In: *Sixth Asia Modelling Symposium*. pp. 200–204 (2012)
20. Malavolta, I., Mostarda, L., Muccini, H., Ever, E., Doddapaneni, K., Gemikonakli, O.: A4wsn: an architecture-driven modelling platform for analysing and developing wsns. *Software & Systems Modeling* **18**(4), 2633–2653 (2019)
21. Minakov, I., Passerone, R., Rizzardi, A., Sicari, S.: A comparative study of recent wireless sensor network simulators. *ACM Transactions on Sensor Networks (TOSN)* **12**(3), 20 (2016)
22. Musznicki, B., Zwierzykowski, P.: Survey of simulators for wireless sensor networks. *Journal of Grid and Distributed Computing* **5**, 23–50 (09 2012)
23. Navarro, D., Mieleveville, F., Galos, M., Carrel, L.: Simulation of hardware and software in heterogeneous wireless sensor network. *Journal on Advances in Networks and Services Volume 7, Number 1 & 2* (2014)
24. Nayyar, A., Singh, R.: A comprehensive review of simulation tools for wireless sensor networks (wsns). *Journal of Wireless Networking and Communications* **5**, 19–47 (2015)
25. Nursetitov, N., Paulson, M., Reynolds, R., Izurieta, C.: Comparison of json and xml data interchange formats: a case study. *Caine* **9**, 157–162 (2009)
26. Papadopoulos, G.Z., Kritsis, K., Gallais, A., Chatzimisios, P., Noel, T.: Performance evaluation methods in ad hoc and wireless sensor networks: a literature study. *IEEE Communications Magazine* **54**(1), 122–128 (2016)

27. Perla, E., Huggard, M., Mc Goldrick, C., Carbajo, R., Cathin, A.: Powertossim z: Realistic energy modelling for wireless sensor network environments. In: Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (2008)
28. Pestic, D., Radivojevic, Z., Cvetanovic, M.: A survey and evaluation of free and open source simulators suitable for teaching courses in wireless sensor networks. In: Internat. Convention on Information and Communic. Tech, Electronics and Microelectronics. pp. 895–900 (2017)
29. Rahman, M., Pakstas, A., Zhigang Wang, F.: Network modelling and simulation tools. Simulation Modelling Practice and Theory **17** (2013)
30. ur Rehman Khana, A., Bilal, S., Othman, M.: A performance comparison of networks simulators for wireless networks. Internat. Conf. on Control System, Computing and Eng. (2012)
31. Robinson, S.: Conceptual modelling for simulation part i: Definition and requirements. Journal of the Operational Research Society **59**, 278–290 (2008)
32. Saginbekov, S., Shakenov, C.: Testing wireless sensor networks with hybrid simulators. arXiv preprint arXiv:1602.01567 (2016)
33. Sarkar, N.I., Halim, S.A.: A review of simulation of telecommunication networks: simulators, classification, comparison, methodologies, and recommendations. Cyber Journals pp. 10–17 (2011)
34. Socolofsky, T.J., Kale, C.J.: Tcp/ip tutorial. Tech. rep. (1991), no. RFC 1180
35. Stetsko, A., Stehlik, M., Matyas, V.: Calibrating and comparing simulators for wireless sensor networks. In: Internat. Conf. on Mobile Ad-Hoc and Sensor Systems. pp. 733–738 (2011). <https://doi.org/10.1109/MASS.2011.80>
36. Sundani, H., Li, H., Devabhaktuni, V., Alam, M., Bhattacharya, P.: Wireless sensor network simulators a survey and comparisons. Internat. Journal of Computer Net. **2**(5), 249–265 (2011)
37. Terrasson, G., Briand, R., Basrour, S., Dupe, V., Arrijurria, O.: Energy model for the design of ultra-low power nodes for wireless sensor networks. Procedia Chemistry **1**, 1195–1198 (2009)
38. Vasanthi, V.: Simulators and emulators used for wireless sensor network. International Journal of Advanced Research in Computer and Communication Engineering **6** (1 2017)
39. Yick, J., M., B., G., D.: Wireless sensor network survey. Computer networks **52:12**, 2292–2330 (2008)
40. Yu, F., Jain, R.: A survey of wireless sensor network simulation tools. Washington University St. Louis, Dep. of Science and Eng. (2011)

Authors

Michel Bakni received the B.S. degree in telecommunication and electronics from Tishreen University, Lattakia, in 2013 and the M.S. degree from the University of Technology of Belfort-Montbéliard (UTBM), France, in 2017, in mobile and distributed networks. He is currently pursuing the Ph.D. degree with the Doctoral School of the University of Bordeaux (UBx) and at ESTIA, a superior engineering School for Advanced Industrial Technologies. His research interests include Simulation, Wireless Sensor Networks, and Energy consumption optimization.

Luis Manuel Moreno is graduated in Telecommunications Engineering at Universidad Simón Bolívar, Venezuela, in 2019. His main areas of research interest are operating systems, distributed systems and embedded systems.

Yudith Cardinale is a Full Professor in Computer Science Department at Universidad Simón Bolívar (USB) since 1996. She graduated with honors in Computer Engineering in 1990 at Universidad Centro-Occidental Lisandro Alvarado, Venezuela. She received her M.Sc. Degree and Ph.D. in Computer Science from USB, Venezuela, in 1993 and 2004 respectively. Her research interests include parallel processing, distributed processing, operating systems, high performance on grid and cloud platforms, and web services composition, including semantic web. She is the Director of the Parallel and Distributed Systems Research Group (GRyDs) at USB and coordinates several national and international research projects. She has written a huge range of publications in areas such as parallel computing, grid computing, parallel check pointing, collaborative frameworks, and Semantic Web.

Guillaume Terrasson received the M.S. degrees in Microelectronics and the Ph.D degrees in Electronics from the University of Bordeaux 1, France, in 2004 and 2008 respectively. The Ph.D. was obtained in collaboration with the ESTIA Engineering School, Bidart, France and the TIMA laboratory, Grenoble, France. Since 2008, he is a researcher at ESTIA RECHERCHE. Qualified in sections CNU in 2010 (French Universities National Council) n63 (electrical engineering, electronics, photonics and systems), his research interests include critical embedded systems applied to aeronautics as well as wireless sensor networks. Guillaume TERRASSON has also managed collaborative European or National programs like Interreg POCTEFA.

Octavian Curea received the Engineer degree in Electrical Engineering at Polytechnic Institute Traian Vuia of Timisoara, Romania, in 1994. He pursued his studies in France by obtaining the M.Sc. degree in 1997, and the Ph.D. degree in 2001, at University of Le Havre, GREAH laboratory. In 2004 he joined ESTIA as associated profesor and he became full profesor in 2017. His research interests include the electrical microgrids from the point of view of energy management, communication networks, power electronics, power quality.

Public-Key Based Authentication Architecture for IoT Devices Using PUF

Haji Akhundov¹, Erik van der Sluis², Said Hamdioui¹, and Mottaqiallah Taouil¹

¹ Delft University of Technology, Delft, The Netherlands

H.Akhundov@tudelft.nl, S.Hamdioui@tudelft.nl, M.Taouil@tudelft.nl,

² Intrinsic ID B.V., Eindhoven, The Netherlands

Erik.van.der.Sluis@intrinsic-id.com

Abstract. Nowadays, Internet of Things (IoT) is a trending topic in the computing world. Notably, IoT devices have strict design requirements and are often referred to as *constrained devices*. Therefore, security techniques and primitives that are lightweight are more suitable for such devices, e.g., Static Random-Access Memory (SRAM) Physical Unclonable Functions (PUFs) and Elliptic Curve Cryptography (ECC). SRAM PUF is an intrinsic security primitive that is seeing widespread adoption in the IoT segment. ECC is a public-key algorithm technique that has been gaining popularity among constrained IoT devices. The popularity is due to using significantly smaller operands when compared to other public-key techniques such as RSA (Rivest Shamir Adleman). This paper shows the design, development, and evaluation of an application-specific secure communication architecture based on SRAM PUF technology and ECC for constrained IoT devices. More specifically, it introduces an Elliptic Curve Diffie-Hellman (ECDH) public-key based cryptographic protocol that utilizes PUF-derived keys as the root-of-trust for silicon authentication. Also, it proposes a design of a modular hardware architecture that supports the protocol. Finally, to analyze the practicality as well as the feasibility of the proposed protocol, we demonstrate the solution by prototyping and verifying a protocol variant on the commercial Xilinx Zynq-7000 APSoC device.

1 Introduction

Secure communication has been paramount throughout history [1]. Although in the early ages it was mainly found in niche applications such as the military and royal society, today it is an inevitable part of our daily lives. The recent rapid proliferation of IoT, a diverse set of devices that are connected to the Internet, imposes new challenges for the designers to keep protecting our privacy, security, and safety [2]. Today, the design of use-case specific solutions and its time-to-market are the biggest challenges in this competitive and rapidly developing IoT semiconductor industry, where developers rely on ad-hoc security solutions. Therefore, there is an urgent need to create cost-effective secure solutions with a short-time development, which are an important facilitator in the IoT market.

Although there is a lot of work published to address the above issues, it mainly focuses on individual aspects such as protocol design or implementations. One of the first works in this field was in 2004 by Lee et al. [3] who showed that individual Integrated Circuits (ICs) can be identified and authenticated using Physical

Unclonable Functions (PUFs). Later, publications such as [4] described a low-cost authentication protocol of individual ICs using PUF. However, that protocol is basic and is not suitable with so-called weak PUFs [5]. More authentication protocols came after that such as in [6], [7], [8] and [9] and a more recent in [10]. Only a few publications that combine both authentication protocols using PUFs and lightweight cost-effective implementations exist in the literature such as [11]; the authors developed a lightweight Application-specific Instruction Set Processor (ASIP) that supports certain existing authentication protocols based on reverse fuzzy extractor (RFE) constructions, rather than reusing existing components. Still, most works typically address the aspects of the authentication protocol but do not address cost-effectiveness and/or time to market aspects. A solution that satisfies all the requirements is still needed.

In this work, we focus on developing a cryptographic protocol based on Elliptic Curve Diffie-Hellman (ECDH) that enables efficient hardware design by reusing readily-available components for an efficient and fast time-to-market design. In that regard, we designed and developed an efficient and cost-effective solution. In short, the contributions of this paper are:

- A *protocol* that enables secure communication between constrained devices and a resource-rich party in *untrusted* fields, and using PUF-derived keys as the root-of-trust. The protocol is based on a conventional ECDH key agreement scheme and a fuzzy extractor using code-offset method for accommodating a PUF.
- A modular *hardware architecture* where the key components are implemented in hardware while minimizing the impact on the silicon footprint. Here we emphasize on the fact that readily-available off-the-shelf components such as the NaCl core [12] can be used to speed up the development cycle. The core can be used to perform elliptic curve scalar multiplications on a patent-free elliptic curve - Curve25519 [13], offering 128 bits of security.
- A proof of concept based on a Zynq board to demonstrate how such a solution can be quickly prototyped using ‘off-the-shelf’ components.

The rest of this paper is organized as follows. Section 2 provides background information. Section 3 discusses protocol design for our intended application; we propose a total of four variants of this protocol, each with its own pros and cons. Section 4 presents a modular hardware architecture design that supports the protocol. Section 5 gives the proof of concept used for validation. Finally, Section 6 provides the conclusion.

2 Background

In this section, we provide a brief background on the working principle of an SRAM-PUF that we use in our work for silicon authentication.

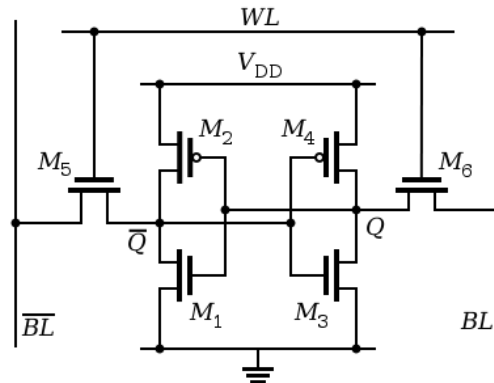


Fig. 1: Conceptual schematic of a 6T SRAM cell

The concept of PUFs was first introduced by Pappu [14] in 2001 as a *hardware security primitive* that can be used for silicon authentication. Later, Maes [15] extended this concept to ‘expression of an inherent and unclonable instance-specific feature of a physical object.’ In essence, PUFs are functions that take *challenges* as an input and generate *responses* that are random but unique for a specific device [16]. Using PUFs, it is possible to create a stable, unique, and device-dependent fingerprint, which can be used as a secret key or a unique device identifier [17], [18]. Therefore, PUFs are applied in several applications such as anti-counterfeiting, device authentication, and hardware/software binding applications [18].

There is a broad taxonomy of PUFs today, such as delay-based arbiter PUFs and ring oscillator PUFs, memory-based SRAM PUFs, and butterfly PUFs [19]. SRAM PUFs are of particular interest to the industry compared to other types of PUFs; Integration of SRAMs in the modern systems do not require special manufacturing techniques since SRAMs can be synthesized using standard cells. Furthermore, they are readily available in most existing systems. SRAM PUF is a memory-based PUF construction that uses intrinsic random start-up cell values to create challenge pair responses [20]. SRAM is a standard cell component that is composed of 6 transistors. Figure 1 shows a typical six transistor SRAM cell design, consisting of two cross-coupled Complementary Metal Oxide Semiconductor (CMOS) inverters using four transistors M_1 through M_4 . Transistors M_5 and M_6 are known as the pass transistors. The wordline (WL), bitline (BL), and its complement are used to access the cell. For performance reasons, the two inverters in the SRAM cell are designed in a well-balanced, symmetrical way. However, the small and random sub-micron process variations in the manufacturing process cause different physical properties of the transistors. These differences in the transistors of the SRAM cell causes a skew. Due to this skew, a cell acquires a preferred state of a logic ‘0’ or a logic ‘1’ when powered on, referred to as one bit of ‘electronic’ fingerprint. This

phenomenon of inherent, device-unique variations makes SRAM PUFs construction possible [19]. It is resistant to cloning even if one can get their hands on the circuit design/layout files since the skew is not visible in the layout. With the current manufacturing process variations are inevitable and cannot be controlled; therefore, cloning an SRAM PUF yields to be tough or even impossible [15]. Every SRAM cell upon power-up can provide one bit of such electronic fingerprint. Hence, arrays of *uninitialized* SRAM cells can be used to identify devices, securely store, and generate cryptographic keys on devices.

In this work, we use SRAM PUF-derived secret key as the hardware root-of-trust in authenticating constrained IoT devices in the field. Note that each time the key must be reliably extracted. Techniques such as *entropy extraction* and *error-correcting codes* are used to achieve this [17].

3 Protocol Design

In this section, we show how a secure protocol can be efficiently composed for an IoT application. First, we describe what confidentiality and authentication are, and then provide a realistic use case. After that, we present the main protocol and variants thereof. Finally, we discuss the advantages and disadvantages of the proposed protocols.

Confidentiality and authentication are some of the core criteria of a secure system [21]. *Confidentiality* is a service used to keep the information accessible only to authorized users. *Authentication* is a service that verifies the identity of users or entities and therefore ensures that its data or the entity can be trusted. In order to achieve confidentiality and authentication between devices, one may employ authenticated encryption techniques. However, before establishing an encrypted channel, communicating parties must share the same key. Due to the key distribution problem [22], key exchange protocols have emerged. Using a key exchange (key-agreement) protocol, involved parties agree on a shared secret key in such a way that all parties influence the outcome, without transferring the actual key itself over an untrusted channel. Key-agreement protocols rely on the exchange of authentic public-key components of the involved parties, i.e., keys that truly belong, therefore prove the identity of claimed users or entities. Next, we present a use case where a key-agreement protocol is used to achieve this.

3.1 Use Case

The use case under consideration in this paper is illustrated in Figure 2; a resource-rich server communicates with constrained IoT devices in an untrusted field. The two fundamental assumptions about this untrusted field are that the communication is susceptible to both *passive and active attacks* because there is very little or no control over it. In a passive attack, an intruder can only eavesdrop on the

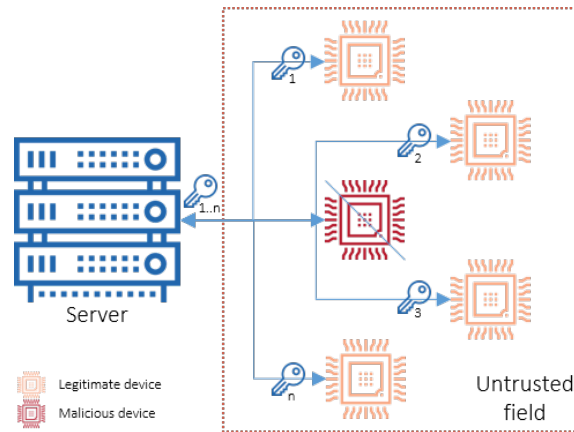


Fig. 2: Use-case scenario

communication. In an active attack, an intruder may also transmit, replay, modify or delete specific communication messages. We see that there are two types of devices: legitimate and malicious. The latter should not be authenticated. The need for *secure communication* in this use case is, therefore, obvious.

3.2 Protocol Description

As pointed out in the background section, establishing secure communication between a device and a server requires a secure protocol. The goal of the protocol is to derive shared keys between the server and legitimate devices in the field, therefore, enabling an authenticated and encrypted communication. We will briefly look at four slightly different scenarios with different requirements that result in protocol modifications and discuss their advantages/disadvantages. In all the scenarios, the communication is to be established between a server and a device enrollment. By using a Trusted Third Party (TTP), we add substantial flexibility due to the Public Key Infrastructure (PKI) and therefore inherit the benefit of certificate management, e.g., device's certificate revocation, etc. The four scenarios are:

- *Scenario 1*: we wish to achieve mutual authentication and certificate management, and use as little Non-volatile Memory (NVM) as possible on the device. To achieve mutual authentication, the server and devices need to be enrolled by a TTP. To reduce NVM requirement and improve certificate management, cloud infrastructure is used. The cloud infrastructure is a database that is meant to be accessible by the TTP and the server for storing and retrieving digital certificates of the enrolled devices, respectively.
- *Scenario 2*: Therefore, similarly as in scenario 1, both parties are enrolled by a TTP; however, cloud infrastructure is not used here.

- *Scenario 3*: we wish to achieve one-way authentication (which could be sufficient in some instances) and certificate management, and again use as little as possible of device’s NVM. To achieve this, only the devices are enrolled by the TTP, and cloud infrastructure is used.
- *Scenario 4*: we loosen up all the constraints, i.e., one-way authentication, no reduced NVM requirement, and no certificate management is required. Clearly, this is the minimal version of the protocol, requiring only the enrollment of the devices and no cloud infrastructure.

For each of the scenarios, we present four different protocol variants, referred to as Protocol A, B, C, and D, respectively. In the rest of the section, we focus on protocol Variant A that satisfies the most demanding Scenario 1 in detail, shown in Protocol 1.1. Other protocol variants are briefly explained afterward.

Variante A: The protocol is divided into two stages. In the first stage, the device is enrolled by a Trusted Third Party (TTP) in a secure environment. This is typically done only once during the life-cycle of a device. The second stage, key-agreement, and authentication, takes place in the field whenever necessary, i.e., communicating parties establish a shared key before communication. The notations, followed by a detailed description of Protocol 1.1 are described below:

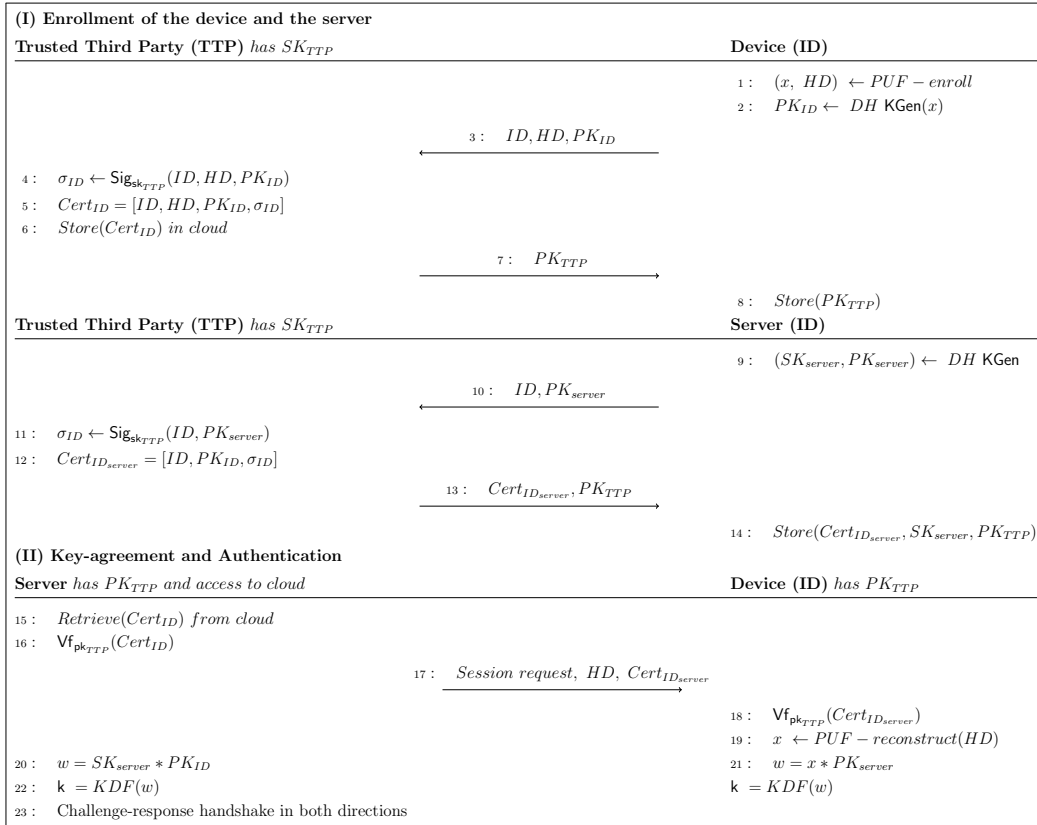
Table 1: Protocol Legend

Symbol	Description	Symbol	Description
ID	Device ID, 48 bits	SK_{TTP}	TTP’s secret key, 256 bits
PK_{TTP}	TTP’s public key, 256 bits	x	PUF-based secret key, 256 bits
HD	Helper data, 752 bytes	PK_{ID}	Device’s Public Key, 256 bits
σ_{ID}	Digital signature, 256 bits	SK_{server}	Server’s secret key, 256 bits
PK_{server}	Server’s public key, 256 bits	w, k	Shared secret and key, 256 bits
$Cert_{ID}$	Device’s Certificate	$Cert_{ID_{server}}$	Certificate
*	Scalar multiplication	$KDF()$	Key Derivation Function
Vf	Signature Verification		

PUF-enroll: generates a (PUF-derived) cryptographically secure key x and helper data HD that is used in the decoding stage of the key reconstruction in the field.

PUF-reconstruct: is the reconstruction process of the secure key x that was enrolled earlier.

DH KGen(x): calculates the public key based on the private key x . In ECC, it corresponds to a scalar multiplication (*) of the scalar x and the base point of a particular elliptic curve as per curve’s specifications.



Protocol 1.1: Protocol based on Diffie-Hellman Key Exchange (DHKE) using Physical Unclonable Function (PUF)-derived key. Variant A - *Achieving mutual-authentication and low Non-volatile Memory (NVM) requirement using the cloud infrastructure.*

$KDF()$: is a Key Derivation Function that should be used to derive a quality secret key from a shared secret [23].

ID : is an identification number unique to a device.

(I) Enrollment All IoT devices must first be enrolled with a TTP before their operation in the field. The enrollment phase happens as follows:

Step 1: Device's PUF response is enrolled, and the key-generation subsystem generates a cryptographic key x , along with Helper Data (HD). The HD is required to reconstruct the private key x from the same PUF device in later stages. This is explained in more detail in Section 4.

Step 2: The generated key x is used to calculate its corresponding public key. This computation is performed using scalar multiplication in a suitable elliptic curve group, the result of which is a point on the curve. Note that in general, depending on the cryptosystem, x may not be directly used as a private key. A post-processing step may be needed.

Step 3: The device sends its identifier ID , HD and the computed public key to the TTP.

Step 4-5: The TTP signs the received data using its private key and generates a certificate.

Step 6: TTP stores the device's certificate to a cloud. The certificate binds the device's ID to its PUF based public key.

Step 7-8: The device must be able to verify the identity of the server it is trying to communicate within the field by verifying its certificate. Therefore, an additional step in device enrollment is sending and storing the TTP's public key PK_{TTP} on the device. To guarantee the security of this protocol, this key must be stored securely on the device, i.e., it cannot be tampered with. Failing to do so allows malicious servers to masquerade as a trusted one.

Step 9-14: In order to achieve mutual authentication, the server needs to be enrolled by a TTP similarly as the device; a digital certificate is issued to the server.

(II) Key-agreement and Authentication In the field, the device must be able to establish a secure and authenticated communication with the server.

Step 15-16: The server retrieves the certificate of the desired device from the cloud and verifies it.

Step 17: The server initiates communication by sending a *session request* message, HD of the device, and the certificate to the device for verification.

Step 18: The device verifies server's certificate using PK_{TTP} .

Step 19: The device reconstructs the PUF-based private key x using the received HD .

Step 20-21: Both parties have the required information to calculate the shared secret w using scalar multiplication.

Step 22: A *Key Derivation Function* is used for privacy-enhancing purposes to derive a cryptographically secure shared key on both sides.

Step 23: A challenge-response handshake in both directions is necessary to make sure that the calculated shared keys on both sides are equivalent.

3.3 Other Variants

Alternatively, if one wishes to have simpler protocol variants (Scenarios 2-4), i.e., no cloud infrastructure or simply only one-way authentication, one may use a simpler protocol. In this subsection, we present three variants with different complexities; similarly to Variant A, all of them are divided into two stages.

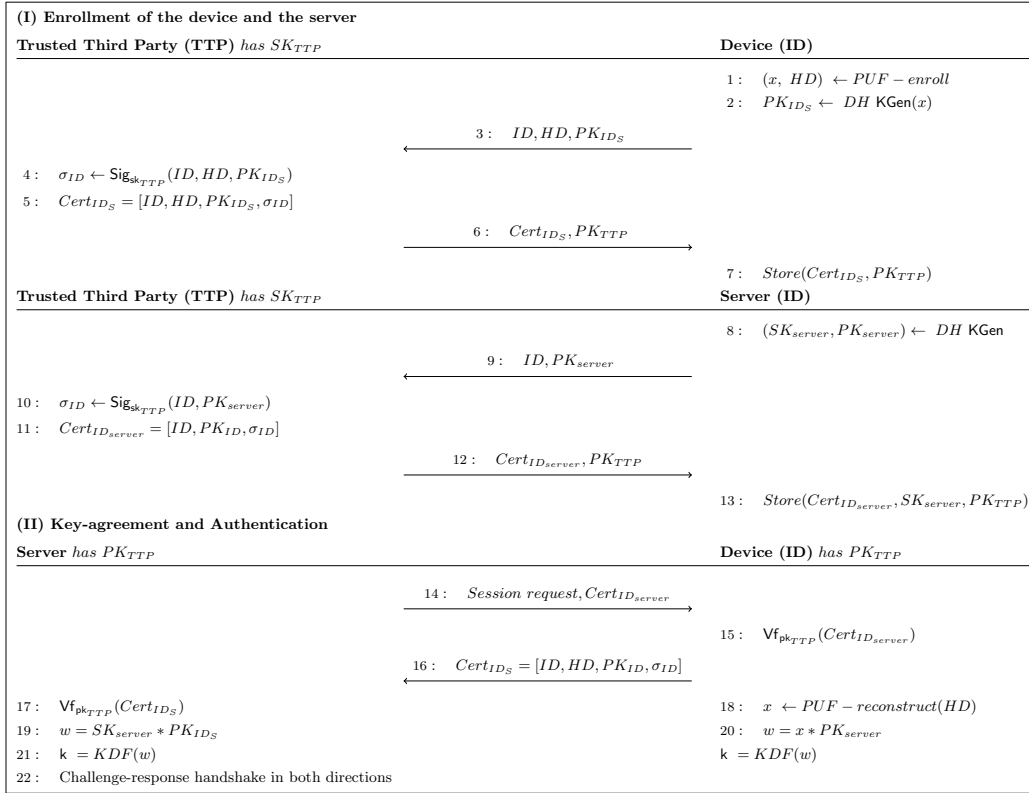
Variant B: This protocol variant, shown in Protocol 1.2, accommodates Scenario 2; therefore, it does not require a cloud infrastructure, while still having mutual authentication. The cloud infrastructure is removed in this variant. The certificate that is generated in Step 5 is transmitted to and stored on the device itself, therefore requiring more NVM storage space. Instead of Step 15, upon a session request, the device sends its previously-stored certificate to the server.

Variant C: This protocol variant, shown in Protocol 1.3, accommodates Scenario 3, therefore, mutual authentication is removed, i.e., Steps 7 - 14 are omitted. This protocol uses cloud infrastructure to reduce the NVM storage requirement on the device and provides certificate management. Step 17 is modified, and Step 18 is removed since no server certificates are involved. The server fetches and verifies the certificate from the cloud, generates an ephemeral DH key-pair, and sends its contribution (PK_{server}) to the device. At this point, both parties can start the key generation process, as seen in Steps 19-23 of Variant A. Note that a challenge-response handshake in one direction is sufficient here.

Variant D: This protocol variant, shown in Protocol 1.4, accommodates Scenario 4, therefore is the simplest variant of the protocols. In this variant, only the device is enrolled, and the certificate that is generated in Step 5 is transmitted to and stored on the device itself. Therefore, Steps 6-16 are omitted. Steps 17-18 are modified as follows; upon a session request, the device sends the certificate back that was stored on the device during enrollment. The server verifies the certificate, generates an ephemeral DH key-pair, and sends its contribution (PK_{server}) to the device. At this point, both parties can start the key generation process, as seen in Steps 19-23 of Variant A. Note that a challenge-response handshake in one direction is sufficient in this case.

3.4 Protocol Evaluation

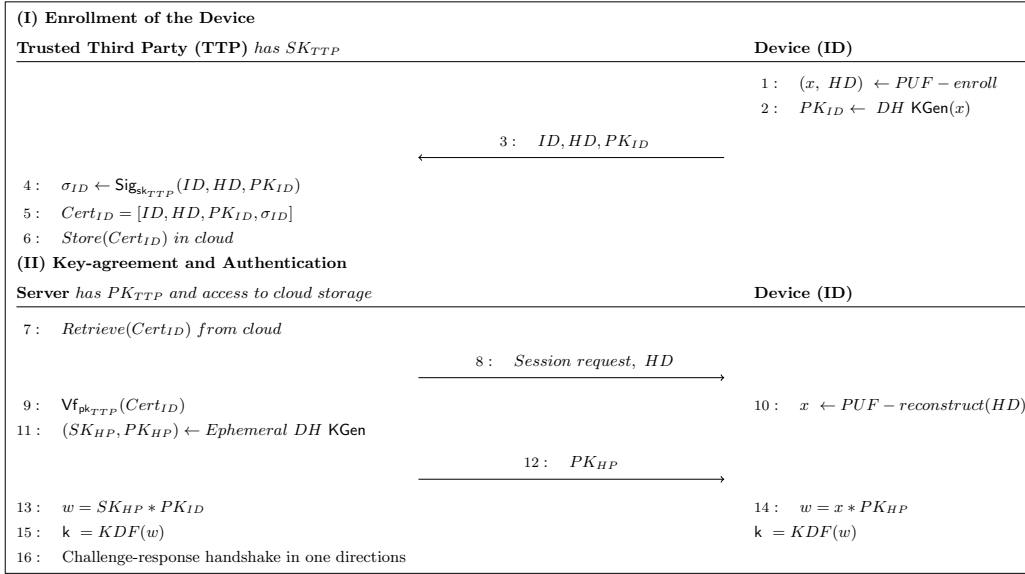
The fundamental security of the proposed protocols relies on the fact that they are built based on the well known ECDH protocol. That said, several assumptions need to be in place to guarantee security. Firstly, we assume that TTP is indeed trusted and that SK_{TTP} is well protected. Secondly, in Protocols A and B, the storage of TTP's public key PK_{TTP} on the device must be secure. Although PK_{TTP}



Protocol 1.2: Protocol based on DHKE using PUF-derived key. Variant B - *Mutual-authentication and no cloud infrastructure.*

is public information, it must not be tampered; if tampered, the security would be compromised, i.e., a malicious server would be able to communicate with the device. Lastly, storing the device certificate $Cert_{ID}$ on the device or in the cloud does not need to be that secure, because $Cert_{ID}$ is public information. Stealing its contents will not give any advantage to anyone. However, modifying it would render it useless, and could be used to perform a denial of service attack on the device, which is why secure storage is still recommended.

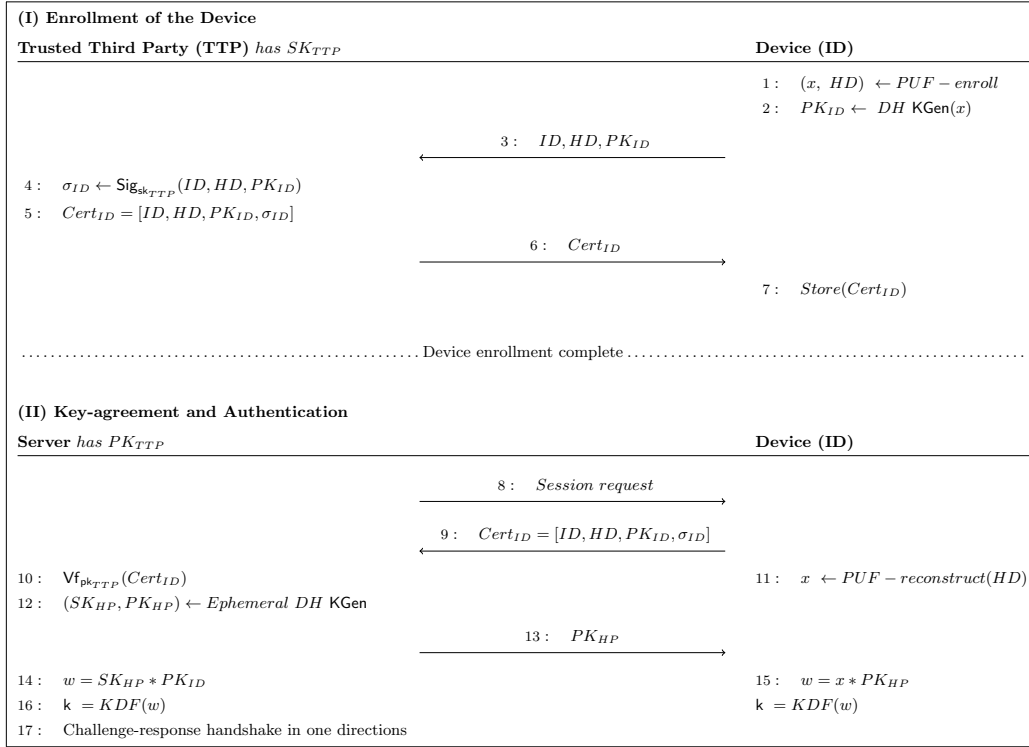
Although the differences between the protocol variants are slight changes, the results in terms of communication performance, functionality, etc. can be significantly different, as we will see. The main properties of the protocol variants A-D are summarized in Table 2. Options/properties with (+) are desired, whereas (-) are not. From the table, we see that in all protocol variants, authenticate the device, whereas only Variants A and B authenticate the server as well. The NVM requirement is discussed later in more detail. Another important point to note is



Protocol 1.3: Protocol based on DHKE using PUF-derived key. Variant C - *Device authentication and using cloud infrastructure.*

that Variants C and D generate ephemeral DH key-pairs, i.e., two same parties will set up new keys for every session, a property that is desired. Alas, protocol Variants A and B do not possess this quality anymore because the signed long term keys are used in the DH handshake. A quick fix for that would be to generate an ephemeral DH key-pair for the session and use the long-term signed key to sign the new ephemeral key. The device will then have to verify this signature. Table 3 shows some basic properties of the protocol variants such as the number of transfers, data transfer size, and the NVM requirement to perform the enrollment, key-agreement, and authentication. Note that the numbers are based on the following realistic considerations. The size of ID is chosen to be identical to the size of a Media Access Control address (IEEE Standard), which is 48 bits. A key with 256 bits of entropy needs approximately 720 bytes of SRAM and 752 bytes of HD based on one of the specific implementations of Intrinsic-ID Quiddikey PUF technology. Moreover, the following analysis is focused more on the device side and the interactions with the device due to its constrained nature.

The *Number of Transfers* shows the number of message transactions during enrollment and in the field. As we can see, Variant B has the most data transfers, whereas Variant C has the least. However, the difference is mainly in the enrollment phase and not during operation; hence, the performance during run-time is similar between them.



Protocol 1.4: Protocol based on DHKE using PUF-derived key. Variant D - *Device authentication and no cloud infrastructure*.

The *Data Transfer Size* shows the size of the messages in bits needed to be communicated during the transactions. This metric is essential for constrained devices because every bit sent consumes power. Interestingly, all protocol variants transfer more or less the same amount of data.

The *NVM Requirement* shows how much data need to be ‘permanently’ stored on the device. Variant B requires the most since it needs to store the certificate as well as the TTP’s public key, whereas Variant C requires no storage.

4 Architecture Design

In the previous section, a key-exchange protocol based on Elliptic Curve Diffie-Hellman (ECDH) and Physical Unclonable Function (PUF)-derived key has been proposed along with several protocol variants. In this section, we design a template of a hardware architecture that enables these protocols on a constrained IoT device. The template contains vital components that are necessary to secure the application. The template enables a quick design of high-level hardware architecture. After

Table 2: Distinguishing Properties of Protocol Variants A-D

Protocol	Device Authentication Server Authentication NVM Requirement	Cloud Infrastructure	Certificate Management	Sig. Verification on Device
Variant A	+ + negligible (+)	required (+/-)	online (+)	required (-)
Variant B	+ + large (-)		offline (+/-)	required (-)
Variant C	+ - none (++)	required (+/-)	online (+)	
Variant D	+ - large (-)		offline (+/-)	

that, we present this high-level system architecture and the components that it is comprised of, followed by a proof-of-concept validation in the next section.

The constrained device must include a minimal set of primitives as elaborated below and shown in Figure 3:

Control unit: A control unit is essential to orchestrate all components and interface with the outside world. For example, the control unit can be a micro-program that implements the protocol and handles the interface to the outside.

ECC Scalar Multiplication unit: The protocols are based on ECDH; hence, scalar multiplication operation $*$ is used on the device both during enrollment and in the field. The scalar multiplication is the most compute-intensive operation in ECC.

PUF System: The protocol is based on PUF-derived keys, i.e., PUF-technology is used for silicon authentication. The selected PUF is an SRAM-PUF due to its availability in most systems. The following are the integral parts of the SRAM-PUF system:

- **SRAM:** For an SRAM PUF, a block of uninitialized SRAM must be available in the system.
- **Fuzzy Extractor:** SRAM start-up values are typically noisy, mostly due to environmental factors, e.g., temperature variation. To compensate for this noise, a fuzzy extractor is used for a reliable and stable secure-key reconstruction [24–26]. A detailed description of a fuzzy extractor, alongside its security or reliability

Table 3: Properties of Protocol Variants A-D

	Variant A	Variant B	Variant C	Variant D
Number of Transfers				
Stage I	4	4	1	2
Stage II	4	5	4	5
Total:	8	9	5	7
Data Transfer Size in bits (with device only)				
Stage I	6576	6832	6320	6576
Stage II	6736	7296	6368	6928
Total:	13312	14128	12688	13504
NVM Requirement (device)	256	6832	0	6576
	{PKttp}	{Certid}	{PKttp}	{Certid}

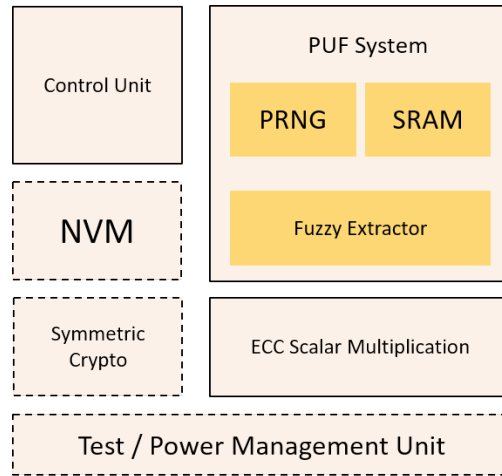


Fig. 3: Conceptual Hardware Architecture

evaluation, is outside the scope of this work. For the sake of proof of concept, we use the simple code offset method, which constitutes of two phases, *enrollment* and *reconstruction*. During enrollment, Helper Data (HD) HD is calculated as $HD = R \oplus C = R \oplus Encode(S)$, where R is the initial PUF response and C is the code, which is the result of an encoded secret key S . During reconstruction, the noisy PUF response R' is read out and the noisy code C' is reconstructed using $C' = R' \oplus HD = R' \oplus (R \oplus Encode(S)) = noise + Encode(S)$. Eventually, the secret key S is obtained by $S = Decode(noise + Encode(S))$.

- **Pseudo Random Number Generator (PRNG):** Essentially, SRAM-PUF is used as a key-storage. During the PUF enrollment stage, a key S must be supplied to the fuzzy-extractor to be programmed. One such source of a key can be a PRNG. PRNG needs to be seeded with a TRNG. For that, the noise in the PUF responses could be used.

The above list represents a minimalist set of components. Optionally, based on the protocol variant the final design might include additional modules such as a *test unit*, *Non-volatile Memory (NVM)*, *symmetric crypto unit*, and a *power management unit*.

5 Proof of Concept

In this section we present the proof of concept that is used for validation as well as a discussion.

In order to demonstrate the practicality of the proposed protocol, we build a prototype using off-the-shelf components that satisfies all requirements. As a prototyping platform, we choose the Xilinx Zynq-7000 family All Programmable System on Chip (APSoC) device. This platform hosts a General Purpose Processor (GPP) and an Field Programmable Gate Array (FPGA). This gives the flexibility to design and develop both hardware, software, and hardware/software co-design paradigms. For this prototype, we selected the *NaCl core* [12] to implement the scalar multiplication, which is one of the key operations in the DHKE protocol.

NaCl Crypto_box in Hardware - which we will refer to as “NaCl core” or simply “NaCl” - is an example of low-resource hardware implementation of the widely known *crypto_box* function of the ‘Networking and Cryptography library’ (NaCl) [12]. The NaCl core is in the public domain, making it worthwhile to use. NaCl uses Curve25519 elliptic curve, which is supported by the popular OpenSSL library and is included in the TLS 1.3 [27]. This is the only low-resource hardware implementation of Curve25519 to our knowledge. The NaCl core supports the X25519 Diffie-Hellman key exchange using Curve25519, the Salsa20 stream cipher, and the Poly1305 message authenticator [12]. The NaCl core is implemented as an Application Specific Instruction Processor (ASIP), with a silicon area utilization of 14.6k gate equivalent. It consumes less than 40uW of power consumption at a 1MHz frequency for a 130nm low-leakage CMOS process technology [12]. There are several reasons why this particular core is chosen. Firstly, it is a *technology independent* hardware implementation targeting highly *resource-constrained* devices i.e., *optimized for area*. Secondly, the VHDL code of the core is in the *public domain* and, therefore, freely available for the public. This allows us to modify it to fit our needs. Moreover, by using the NaCl core, we build on top of previous academic work and reduce development time.

The performance of the NaCl core mainly depends on the configuration of the multiplier. The fastest two-cycle version of the core utilizes 2754 LUT Slices on a Xilinx Artix[®]-7 FPGA and takes approximately 830882 cycles for scalar multiplication. However, in order to have the smallest possible area utilization, we configure the NaCl core to use a 16 cycle multiplier at the cost of time [12]. Furthermore, the original core contains other functionalities such as the XSalsa20 and Poly1308 code [12], compiled and stored in the ROM program that we do not need for this

The screenshot displays a web-based interface for managing a device. At the top left, a dropdown menu is set to 'COM6' under the heading 'Serial Port'. To the right, several fields are populated with hexadecimal values: 'Device ID: 4242F4242F', 'AC: AFA00AFA00', 'PK: 05228913D8E479866F1F37E4907606ED98046BB4BEDB287EAE2D8707C12C344C', and 'Share Secret: 04-33-8C-C7-F1-FF-72-54-49-76-C3-40-62-81-94-AE-28-DC-8F-BB-BB-E7-B4-58-49-D1-80-5C-9D-60-15-38'. Below these fields are two buttons: 'Enroll' (orange) and 'Authenticate' (blue). An 'Enter Input' field is present below the buttons. At the bottom, a 'Device Output' section shows a log of the authentication process, including the device ID, AC, PK, and a shared key.

Fig. 4: Server GUI; used to *enroll*, *authenticate* and provide the output of the device.

work. By reducing the program to its minimum, we further reduced the ROM size by approximately a factor of two. By doing all this, the final NaCl configuration takes 3.475.123 cycles and has the lowest area utilization of 946 LUT Slices on our prototyping platform.

The Xilinx Zynq APSoC platform tightly couples a processor together with the fabric, and the communication is possible via the AXI-peripheral. In order to integrate the NaCl core into our prototype setup, we must first wrap the NaCl core into an AXI-peripheral. Secondly, create *hardware interface drivers* to abstract the hardware and expose only the high-level operations to the programmer.

5.1 Validation

The essential components needed for all the protocol variants are the same. Therefore, to validate the protocols, we choose to emulate Protocol D on the platform described above, due to its simplicity. We used a server to perform the key-agreement authentication protocol without the device. To simplify the verification process, a GUI has been developed that executes the protocol steps. A screenshot of this GUI is provided in Figure 4. At the end of the emulation, we verified that both parties (i.e. device and server) derived the same key. Furthermore, in order to emulate malicious behavior in our prototyping setup, we replaced the authentic SRAM by one that has not been enrolled. Our experiment showed that we could identify malicious devices and deny their access.

5.2 Discussion

In this section, we discuss how this solution satisfies the requirements, as well as its limitations.

Mutual authentication key-agreement protocol - One of the requirements was to enable secure communication between resource-constrained and unconstrained IoT devices and provide mutual authentication. Moreover, by using PUF derived keys on the IoT devices, we authenticate silicon. Hence, an Elliptic Curve Diffie-Hellman (ECDH) based protocol was designed using the PUF-derived key. The protocol achieves mutual authentication. Furthermore, by using a cloud infrastructure, we achieve certificate management that can be used to blacklist devices easily. Alternatively, based on a particular need, one can choose from three additional protocol variants.

Fast time to market - We present a generic hardware architecture template that can be used in conjunction with the proposed protocols. The critical component in the entire system is scalar multiplication. We show that an off-the-shelf core such as NaCl can be quickly used for such purposes. Furthermore, this has the potential to be a viable option for a low to medium production, and a fast time to market, which is crucial in the IoT market. Alternatively, if a higher production volume is required, we might need to design an ASIC. Although the non-recurring costs are known to be high for such design, the resulting per-unit price can be substantially minimized this way. Furthermore, ASIC design can be optimized for an area, resulting in the smallest form factor.

Minimizing silicon footprint - To minimize the footprint, we use ECC. Due to its shorter operands, area utilization is significantly reduced when compared to other public-key crypto cores, while achieving the same level of security. Therefore, ECC has been gaining popularity in the community as a suitable candidate for constrained devices. Furthermore, to minimize the silicon footprint of our design, we choose to use SRAM-PUF. SRAM is already present in most systems, and SRAM can be made from standards components. Furthermore, SRAM-PUF technology itself is gaining popularity, and its widespread adoption is imminent.

Validation - In this work, we introduced a systematic approach to building a prototype for the validation of our proposed protocol. The prototype allowed us to verify, evaluate, and analyze the feasibility of such a system.

6 Conclusion

In this paper we proposed a mutual authenticating key-agreement protocol that is based on ECDH which uses a SRAM-PUF based key. The proposed protocol is designed for IoT devices that work under stringent constraints, i.e., area, cost and power. We chose ECDH since ECC is a suitable candidate for constrained devices due to the shorter operand size. Also, SRAM-PUF is chosen to be used due to its expected adoption in IoT. In our base protocol, communicating parties are enrolled by a TTP to achieve mutual authentication with a negligible NVM requirement on the device's side. Furthermore, we use cloud infrastructure to make certificate management possible. In order to comply with different scenarios, three additional

variants of the original protocol are proposed. We further provide a comparison of the variants, showing the trade-offs related to security versus implementation requirements. As future work, we expect to carry out formal security verification of the proposed protocol and its variants. The designed protocol served as a roadmap in drafting a modular hardware architecture. This architecture was prototyped, verified, and its feasibility and practicality were demonstrated on a Xilinx Zynq-7000 APSoC device.

References

1. S. Singh, *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*, 1st ed. New York, NY, USA: Doubleday, 1999.
2. A. Gerber, *Top 10 IoT security challenges*, 2017 (accessed December, 2018). [Online]. Available: <https://developer.ibm.com/articles/iot-top-10-iot-security-challenges/>
3. J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*, 2004, pp. 176–179. [Online]. Available: <http://ieeexplore.ieee.org/document/1346548/>
4. G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings - Design Automation Conference*, 2007, pp. 9–14.
5. C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, aug 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6823677/>
6. K. B. Frikken, M. Blanton, and M. J. Atallah, "Robust Authentication Using Physically Unclonable Functions," in *Information Security*, P. Samarati, M. Yung, F. Martinelli, and C. A. Ardagna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 262–277.
7. A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Reverse fuzzy extractors: Enabling lightweight mutual authentication for puf-enabled rfids," in *Financial Cryptography and Data Security*, A. D. Keromytis, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 374–389.
8. R. Maes, *PUF-Based Entity Identification and Authentication*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 117–141. [Online]. Available: https://doi.org/10.1007/978-3-642-41395-7_5
9. A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung, "End-to-end design of a puf-based privacy preserving authentication protocol," in *Cryptographic Hardware and Embedded Systems – CHES 2015*, T. Güneysu and H. Handschuh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 556–576.
10. M. Barbareschi, A. De Benedictis, and N. Mazzocca, "A PUF-based hardware mutual authentication protocol," *Journal of Parallel and Distributed Computing*, vol. 119, pp. 107–120, 2018. [Online]. Available: <https://doi.org/10.1016/j.jpdc.2018.04.007>
11. A. Aysu, E. Gulcan, D. Moriyama, and P. Schaumont, "Compact and low-power ASIP design for lightweight PUF-based authentication protocols," *IET Information Security*, vol. 10, no. 5, pp. 232–241, 2016. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-ifs.2015.0401>
12. M. Hutter, J. Schilling, P. Schwabe, and W. Wieser, *NaCl's Crypto-box in Hardware*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 81–101.
13. D. J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records," in *Public Key Cryptography - PKC 2006*, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 207–228.

14. P. S. Ravikanth, "Physical one-way functions," Ph.D. dissertation, Cambridge, MA, USA, 2001, aAI0803255.
15. R. Maes, *Physically Unclonable Functions - Constructions, Properties and Applications*. Springer, 2013. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-41395-7>
16. R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*. Springer, 2010, pp. 3–37.
17. R. Maes, V. van der Leest, E. van der Sluis, and F. Willems, "Secure key generation from biased pufs," Cryptology ePrint Archive, Report 2015/583, 2015, <http://eprint.iacr.org/>.
18. V. van der Leest and P. Tuyls, "Anti-counterfeiting with hardware intrinsic security," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, March 2013, pp. 1137–1142.
19. R. Maes and I. Verbauwhede, *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 3–37. [Online]. Available: <https://doi.org/10.1007/978-3-642-14452-3.1>
20. A. Cortez, A. Dargar, G. Schrijen, and S. Hamdioui, "Modeling sram start-up behavior for physical unclonable functions," in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, Austin, USA, October 2012, pp. 1–6.
21. S. S. Kumar, "Elliptic curve cryptography for constrained devices," Ph.D. dissertation, Ruhr University Bochum, 2006.
22. C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, 1st ed. Springer Publishing Company, Incorporated, 2009.
23. H. Krawczyk, "Cryptographic extraction and key derivation: The hkdf scheme," in *Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 631–648.
24. X. Boyen, "Reusable cryptographic fuzzy extractors," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, ser. CCS '04. New York, NY, USA: ACM, 2004, pp. 82–91. [Online]. Available: <http://doi.acm.org/10.1145/1030083.1030096>
25. Y. Dodis, L. Reyzin, and A. Smith, *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 523–540.
26. J.-P. Linnartz and P. Tuyls, *New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 393–402.
27. "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force, Standard, accessed on 05-12.2018.

CUSTOMIZED GARMENT FASHION RECOMMENDATION SYSTEM USING DATA MINING TECHNIQUES

Shukla Sharma¹²³, Ludovic Koehl¹², Pascal Bruniaux¹², Xianyi Zeng¹²

¹GEMTEX

²ENSAIT

³ECOLE CENTRALE DE LILLE

Lille, France

ABSTRACT

Many fashion firms have enabled their business model to give extremely personalized experiences to their customers by using advanced CAD tools like CLO 3D, Marvelous-Designer, Browzwear, Lectra and many more for designing the garment and build a 3D avatar for the customized garment as well as web-based services to be integrated with the web and mobile-based applications. Due to the integration of highly advanced technologies for designing and giving personalized experience has increased the customer's expectations. In this paper, we have presented our initial work to build a garment fashion recommendation system for customized garments, which can be used with mobile and web applications. The proposed system structure is designed on the user's biometric profile and historical data of product order. We have collected the user's historical data from a fashion company dealing with customized made-to-measure garments. Proposed architecture for recommendation system is based on different data mining techniques like clustering, classification and association mining.

KEYWORDS

Recommendation System, BIRCH, Adaptive Random Forest, Incremental learning, data mining, Association mining

1. INTRODUCTION

Digitalization in the fashion industry has automated each step by using CAD based tools as well as web-based platforms and made it easy to build the digital supply chain. Due to the digitalization fashion brands has generated a good amount of information to understand the consumers at various level and opened the door for the adoption of data-driven services based on machine learning-based data analysis. Recently -commerce platform Zalando has opted for data-driven marketing techniques based on artificial intelligence and for the accomplishment of building data-driven services has acquired AI start-ups to work on their business domain [1].

Big market players Amazon, Alibaba, Flipkart, Myntra have also invested to build a strong digital supply chain to grab more customers and fulfill their demand by understanding them with deep

analysis using Artificial Intelligence based services [1]. Deloitte analysis for fashion companies has shown that consumers are willing to bear the expensive amount for highly personalized garments and fashion accessories [2]. Highly personalized garments need a close connection between the consumer and designers and to build the recommendation system to fulfill the gap by considering the customized garment data available recommendation filtering techniques analyzed. The selection of the filtering technique depends on the domain of business and type of available data attributes [25]. Broadly RS can be divided into three categories first is content-based, second is collaborative and third is hybrid filtering techniques. Content-based (CB) filtering technique prepares the recommendation by matching the similarity with the content of the user's preferable items. Also, CB filtering method matches the content of the user's profile by considering attributes that show the interest of users. In the literature of fashion recommendation systems [3] it can be seen that the content-based analytic tools used to know the fashion brand's sales activity. Content-based filtering is not suitable where content for product and user profile is not well-formed. The content-based system fails to provide recommendations by considering product popularity. On the other hand, collaborative filtering (CF) builds the recommendation result by knowing the relationship between the users and items by considering item ratings. Collaborative filtering further can be divided into two types, first is item-based and second is user-based CF. User-based CF works on user rating given to the item and matching corresponding user profiles. The user-based approach becomes insufficient when data sparsity is seen. A small number of rated items between user profiles causes to generate unreliable and poor recommendations. Another issue is the user's profile information gets updated frequently and it requires the recompilation of user-based models. Also, an expectation to have ratings or item interaction data for every item leads CF-based systems to face item cold-start problems.

Another popular Item-based CF was introduced by Amazon to overcome the issues faced in user-based filtering techniques. The item-based filtering technique prepares prediction by calculating the similarity between items. The similarity between items remains stable as compare to user profile similarity and less requirement to recompile the item-based model. The item-based system builds recommendations for users by matching the similarity between the items and items purchased together with the large user population [4]. A collaborative filtering technique with visually explainable recommendations in the fashion industry has been proposed in the literature of recommendation systems.[5]. Item-based CF technique also inherits cold start problems when there is sufficient data to analyze the user's historical behavior for new item sets.

Hybrid filtering(HF) is a combination of various recommendation techniques to overcome the shortcomings of previously introduced Content-based and collaborative filtering. Applying a combination of machine learning algorithms or data mining techniques with a combined approach which is followed in content-based and collaborative filtering can improve the accuracy of the recommendation model. Because the issue of one algorithm can be solved by the other algorithm. In the literature of fashion recommendation systems, hybrid recommendation system is used to find the latest fashion trends [6]. We used a hybrid recommendation system using data mining techniques to analyze the customized shirt's data. Further paper is organized as follows. The next section 2 explains and highlights the existing fashion industrial report and recent issues faced by fashion e-commerce platforms regarding the design of garment. Section 3 we have shown data mining steps with a brief introduction of incremental clustering algorithms. Further sub-section under section 3 shows details about the BIRCH clustering, Adaptive Random Forest classification and association mining algorithms and their results. The last section concludes the activities completed in this research paper.

2. PROBLEM STATEMENT FOR CUSTOMIZED GARMENT RECOMMENDATION SYSTEM

A significant increase and adoption of the CAD tools have given a very closed personalized experience to consumers. Fashion companies are using the latest technologies like virtual AR VR platforms, Mirrors.

A virtual dressing room by GAP provides customers virtual garments over 3D avatars of the user's body morphology. CAD tools helped to design garment designs suited to different body type and 3D visualization made it easier to see the garment's ease over the body. Famous CAD tools like Lectra, CIO3D are used by designers to create the garment pattern and then evaluate the fitting on 3D avatar. Despite having so much detailed information and virtual dressing rooms still, there is a gap between the designers and users' understanding related to fashion attributes. The famous sales drop issue in Zalando has shown that returns are not only because of fittings but also related to garment design and fabric choice for a garment. Stacia Carr Zalando's Director of Engineering mentioned that design and fabric also can have an impact on the returns. Zalando raised alarm to one popular denim brand that faced sales drop due to a small change in the design [7]. In this research paper, to build a fashion recommendation system using each and every attribute of the customized shirt like collar cuff, pocket, fabric, color and biometric profile of the user. Hybrid filtering with data mining techniques is used to build the system. Because the available dataset doesn't fulfill the requirements of content-based and collaborative filtering techniques. Both the filtering techniques needs product description and rating data and rating data collection for a customized garment is not feasible due to the multiple numbers of style attributes of the garment.

Asking users for rating all visible attributes is the complex and not user-friendly approach, for example, shirt attribute collar can have many different types of collar styles and the same applies to other style attributes like shirt button type, pocket type, cuff type, back yoke type. A web platform for customized garments works more closely with complex body measurement techniques because the individual body has its own specificity[8]. Through our research work, we aim to explore different scenario for a customized garment from designer's and consumer's point of view and we contribute as follows:

- Build a close connection between users and designers by analyzing consumer's selection choices for different styles and attributes available on the system.
- GFRS is proposed to give a real-time recommendation by handling incremental data.

Following steps for building hybrid recommendation systems are as follows:

- Recommendation model composition starts from data partitioning by taking the vector of biometric parameters specific to customized garments.
- Data partitioning divides data into small segments each segment contains similar biometric profiles of users.
- Each segment is further classified corresponding to the fitting type.
- Frequent itemset patterns mined by applying association learning to the different variants of a shirt. Frequent itemsets are the combination of attributes selected together for a shirt.

3. METHODOLOGY

Data mining is the process of finding the hidden patterns from large data sets and generate useful insights for business. The development of a recommendation system is specific to. Every domain has different needs of recommendation systems like a proposed system in [9] has focused on fashion company which sells products through online web platforms as well as with offline showrooms. Therefore building the recommendation system with a conventional model used in other e-commerce websites and social networks can not fulfill the requirement of the different types of fashion business models.

3.1. Data Mining

Data mining is not a new term and drills down data to get useful insights for business have a long history. The extraction of the hidden pattern accelerates the pace of decision-makers. The exponential use of latest technologies and the proliferation of data in the fashion industry has raised the need for using big data analysis techniques to get valuable insights for fashion companies [10].

Commonly used clustering algorithms are categorized as partitioning, hierarchical, grid-based, and model-based algorithm [11] Partitioning methods require the value of k for algorithm by a user and it relocates the instances from one cluster to another. K-mean is most commonly used in the algorithm. k-mean partitions data with their respective centroids. The centroid is calculated by taking the mean of all instances in the cluster. K-medoids is another partitioning algorithm also known as PAM (Partition around medoids) Hierarchical based clustering algorithms are further categorized into two types:

- Agglomerative hierarchical clustering follows a bottom-up approach. Initially, each element is considered as a cluster of its own. Every iteration works to combine most similar instances into a bigger cluster node. The iteration process works until the desired cluster structure is formed.
- Divisive hierarchical clustering - follows a top-down approach. All instances are considered as a single cluster at first step and then moving forward with each iteration most heterogeneous cluster of instances divided into sub-clusters and it continues until every object is assigned to their cluster.

Grid-based clustering algorithm CLIQUE is a grid based cluster to work with high dimensional data. It helps to find the cluster in the subspace of high dimensional data and it does not require to select the subspace which might have a cluster. Because of automatic subspace clustering, CLIQUE has been recommended to use in data mining applications[12].

Model-based clustering algorithms use a different model for each cluster and tries to find the best fitting for that model. Statistical learning and neural-based learning are two types of model-based learning. Statistical learning includes COBWEB, GMM, and neural-based learning includes SOM, ART [13]. In our research work, we have clustered data using BIRCH (balanced iterative reducing and clustering using hierarchies) algorithm.

BIRCH algorithm is an unsupervised hierarchical algorithm for clustering. BIRCH is suitable for handling large data sets and incrementally and dynamically handle new metric data points [14]. The BIRCH clustering algorithm completes the process in two steps. The first step is building CF Tree and loads data into a cluster feature tree(CF Tree).CF Tree is a compressed form of data. BIRCH algorithm becomes highly efficient by using summary statistics for minimizing large data sets. CF Tree is built with CFs and each CF is composed of three summary statistics [14]:

- The count represents the number of data values in the cluster.
- The linear sum is the sum of individual coordinates and helps to measure the location of the cluster.
- Squared Sum is the sum of squared coordinates and helps to measure the spread of the cluster.

The second step is clustering the sub-clusters. After the creation of the CF Tree existing clustering algorithm on CF Tree Leaf nodes(sub-clusters) is applied to combine sub-cluster into clusters.

3.2. Dataset Description and Recommendation Model Building Steps

We have collected data from European fashion companies that are dedicated to creating a customized garment. We have considered customized shirts dataset for building an initial model. Detailed description related to data can be seen in Table 1, Table 2, Table 3 GFRS model is designed by combining three steps:

Table 1. Customized shirt data

Month count	Order Count	User Count	Remake Request Count
10	5291	4712	493

Table 2. Customized shirt design attributes

Fabric	Fit	Collar	Placket	Cuff	Pocket	CollarWhite	CuffWhite
331	8	44	14	32	10	2	2

Table 3. user biometric profile

Height	Weight	CollarSize	Age

Data Clustering: Data clustering is used as a pre-processing step, Which becomes useful at later steps of data analysis. Also makes easier to build initial overview on the dataset by applying statistical analysis and machine learning algorithms [15] BIRCH algorithm is used to create the separate groups which are based on user biometric profile for the shirt. Three-dimensional input vector [height, weight, collar size] has passed to get the data in homogeneous segments for further analysis. In this step we have created segments of similar objects for our recommendation model. The reason for doing segmentation on the basis of biometric profile is useful to deal with the extremely different and similar biometric profile of users for example extremely long height

and weight user profiles won't be suitable for group of users who are having extremely short height and weight. Following steps used for the completion of clustering:

- We have used the Scikit machine learning library to implement the BIRCH algorithm for online clustering.
- BIRCH model is trained in 5 iterations where each iteration is containing 1000 approx records to analyze the incremental behavior of the model.
- Silhouette Coefficient, Davies-Bouldin score metrics are used to evaluate the clusters in each iteration.

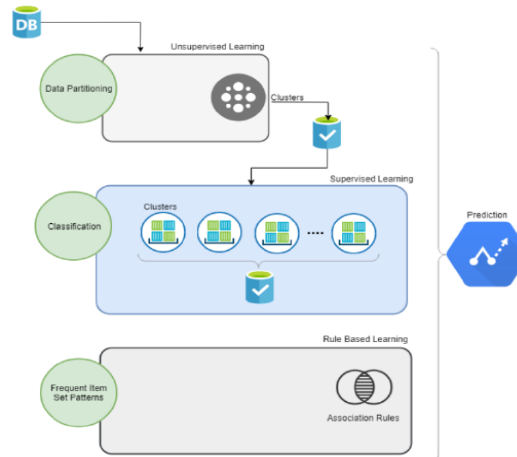


Figure 1. Initial architecture of our Garment Fashion Recommendation System.

Cluster evaluation Silhouette Coefficient metric is used to know how well clusters are formed. It is calculated using mean intra-cluster distance and mean nearest cluster distance [16]. Its value ranges from -1 to 1. Coefficient value 1 is considered as the best value and indicates the good clusters formed. If the coefficient value is going towards 0 it means that formed clusters are overlapping and if the value is -1 then this negative value indicates that samples have been assigned to the wrong cluster or clusters are not well-formed [17]. Davies-Bouldin score's minimum value is 0 which is considered to be good and it is calculated as the average similarity score of each cluster with its most similar cluster.

A similarity measure is calculated as the ratio between within-cluster distances and between cluster distances [17]. Table 4 shows the Silhouette Coefficient metrics increasing as new data is used to train the model cluster formation is reducing the overlap between the clusters. The silhouette score moving towards 1 is considered good and 0 indicates that clusters are overlapping. A negative score means the wrong assignment of samples to the cluster.

Table 4. Cluster evaluation

Iteration	silhouette_coefficient	davies_bouldin_score
0	0.327517934551341	0.98885472069335
1	0.322808536550035	0.80856055643328
2	0.294895935082953	1.01848641365009
3	0.311195152561083	1.02506770111566
4	0.307172835023094	1.04154161015915

Davies Bouldin score is the average similarity score of each cluster with its most similar cluster. decreasing and going towards 0 shows clusters are not similar and less dispersed.

Data Classification: We have used supervised learning in this step using Adaptive random forest classifier to predict best fitting type class for input vector [X, Y] where X containing three biometric parameters [height, weight, collarsize] Y contains fitting type as a label. The adaptive random forest (ARF) classifier is suitable for online learning or to follow an incremental approach. Traditional classifier needs to be retrained if new data comes by losing previous information or appending new data with an older dataset and retrain the model to use all information. Bagging, Random forests are an example of ensemble classifiers. Bagging predicts by aggregating the multiple version of predictors [18]. Random forest is another popular and widely used ensemble classifier that uses a forest of trees and they vote for the popular class [19]. Increase the performance by combining the prediction of all models. ARF is an updated version of random Forest Following characteristics from experiments done in a research paper[20]:

- ARF obtains good classification performance on real-world data.
- A large number of feature handling using a small number of trees.
- ARF can train its base trees in parallel without affecting its classification performance. This is an implementation concern, but it is useful for investigating scalability.
- ARF might not be able to improve on data sets where all features are necessary to build a reasonable model.

Table 5. Classification report for Cluster 1

class_name	f1score	precision	recall	support
3	0,078431373	1	0,040816327	49
4	0	0	0	1
7	0,717557252	0,598726115	0,895238095	210
8	0,32	0,5	0,235294118	102
9	0	0	0	2
accuracy	0,587912088	0,587912088	0,587912088	0,587912088
macro avg	0,223197725	0,419745223	0,234269708	364
weighted avg	0,514203737	0,620144187	0,587912088	364

Table 6. Classification report for Cluster 2

class_name	f1score	precision	recall	support
1	0	0	0	3
3	0,795275591	0,677852349	0,961904762	105
4	0	0	0	3
5	0	0	0	1
7	0,383561644	0,736842105	0,259259259	54
8	0	0	0	2
accuracy	0,68452381	0,68452381	0,68452381	0,68452381
macro avg	0,196472872	0,235782409	0,203527337	168
weighted avg	0,620334915	0,660499823	0,68452381	168

Table 7. Classification report for Cluster 3

class_name	f1score	precision	recall	support
1	0,425531915	0,5	0,37037037	27
3	0	0	0	12
4	0	0	0	3
5	0,4	1	0,25	12
7	0,627345845	0,527027027	0,774834437	151
8	0,401869159	0,494252874	0,338582677	127
accuracy	0,521084337	0,521084337	0,521084337	0,521084337
macro avg	0,309124486	0,420213317	0,288964581	332
weighted avg	0,488120384	0,505575892	0,521084337	332

Table 8. Classification report for Cluster 4

class_name	f1score	precision	recall	support
1	0,476190476	0,5	0,454545455	11
3	0,171428571	0,666666667	0,098360656	61
4	0,133333333	0,5	0,076923077	13
5	0	0	0	5
7	0,653846154	0,488505747	0,988372093	86
8	0,1	1	0,052631579	19
9	0	0	0	1
accuracy	0,5	0,5	0,5	0,5
macro avg	0,219256934	0,450738916	0,238690408	196
weighted avg	0,38550684	0,579990617	0,5	196

Classification Evaluation: In the second step, we trained 4 classifiers based on each group identified in the clustering step. The classification report generated to see the quality of the model's prediction. Classification model evaluation values corresponding to each metric can be seen in Table 5, Table 6, Table 7, Table 8. F1 score, precision, recall, support metrics are used to know the model's prediction over different classes.. Precision shows the classifier's ability to not label positive instance as negative or percentage of prediction was correct. Precision value is calculated for each class by dividing true positives by the sum of true and false positives.

$$tp/(tp + f p) \quad (1)$$

A recall is the percentage of positive cases in the classifier model and for each class, the ratio is calculated by dividing true positive by sum of true positives and false negatives.

$$tp/(tp + f n) \quad (2)$$

Support shows the number of actual occurrences of the class In the specified dataset.

Classification report for some classes f1score value is 0 which gives an indication true positive + false positive == 0 or true positive + false negative == 0 .Also classes showing 0 value for precision means that there were no true positive case found for that class.

Prequential Evaluation: Interleaved test then train is used to test the incremental Adaptive Random Forest Classifier’s accuracy with different pre-trained instances Figure 1 shows mean performance accuracy of classifier increased gradually as the number of instances increased to pre - trained the model.

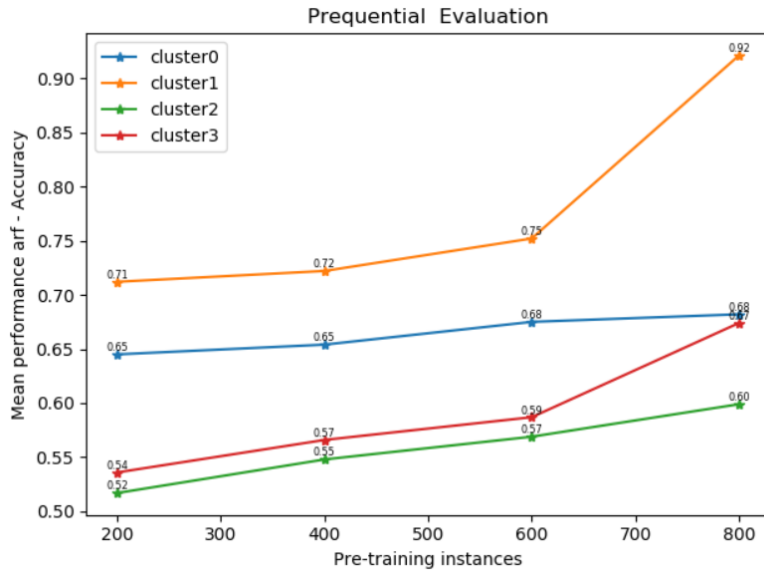


Figure 1. ARF performance accuracy calculation using prequential evaluation

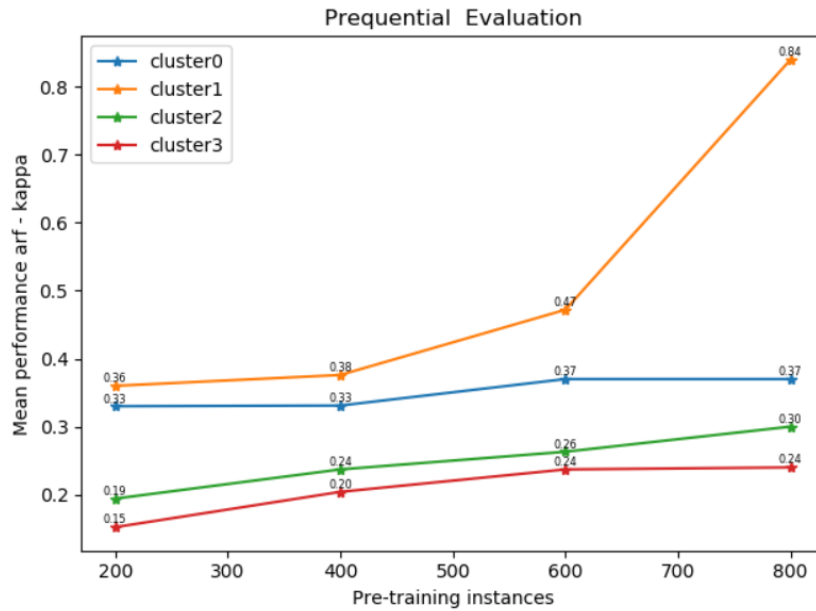


Figure 2. ARF kappa score calculation using prequential evaluation

Figure 2. Shows the kappa score for ARF classifier in a moderate range. Kappa score ranges between -1 to 1 and used to measure inter-annotator agreement, negative score and score close to zero shows chance agreement and 1 means complete agreement.

Association Mining: Association mining has used in the third step to getting the frequent itemset and association rules for the shirt attributes. In the literature of the recommendation system association mining is considered for building a web-based personalized recommendation system where explicit feedback is missing or can not be collected from users [21]. Also, association mining has been used in previous research work for improving the web-based application's performance by getting the best set of rules using the Apriori algorithm [22]. Association mining process needs frequent itemsets to build rules and Apriori is one of the famous widely used algorithms. It was first introduced in 1996 [23] for finding all frequent itemsets using the Apriori candidate generation procedure and next is to get the support of frequent itemsets is counted. Apriori's candidate generation process becomes costly when there are long patterns. On the other hand, FP tree algorithm is an efficient algorithm to find frequent itemsets for long patterns because of its compact tree data structure which helps to avoid repeated data scans [24]. FP Growth is faster than the Apriori algorithm.

FP Growth with minimum support value 0.1 has trained with all set of attributes Fabric, Collar, Cuff, Placket, Pocket, CollarWhite, CuffWhite to get the frequent pattern sets. Getting a number of frequent pattern set depends on the value of minimum support as value goes down a number of frequent itemsets increases. Support, confidence, and lift metrics have been used to filter interesting rules. Association rules are represented as $A \rightarrow C$, where A is antecedent itemset and C is consequent itemset. Support is the combined metric for antecedent itemset and consequent itemset and gives a percentage of transactions that contain both antecedent and consequent. The confidence value for rule is to know the truthness of consequent occurrence corresponding to antecedent in transaction database. Lift value greater than 1 indicates that antecedent and consequents are dependent and if the value is 1 two sets are independent of each other and can't be considered as potential rules.

Table 6. Filtered association rules

antecedents	consequents	support	confidence	lift
{'no_collarwhite',Classic Point_collar}}	{'no_remake'}}	0.156374502	0.945783133	1.02544953
{'Classic Point_collar',no_remake}}	{'no_collarwhite'}}	0.156374502	0.951515152	1.016299162
{'no_collarwhite',Double inc. Cufflinks_cuff}}	{'no_remake'}}	0.230079681	0.950617284	1.030690878
{'Double inc. Cufflinks_cuff',no_remake}}	{'no_collarwhite'}}	0.230079681	0.954545455	1.019535783
{'no_collarwhite',Hai Cutaway_collar}}	{'no_remake'}}	0.120517928	0.930769231	1.009170959
{'no_remake',Hai Cutaway_collar}}	{'no_collarwhite'}}	0.120517928	0.952755906	1.017624393
{'Round Single_cuff',no_remake}}	{'no_collarwhite'}}	0.310756972	0.939759036	1.00374263
{'Italian Semi-Spread_collar',no_remake}}	{'no_collarwhite'}}	0.162350598	0.942196532	1.006346083

Machine learning library: scikit-multiflow open-source framework for machine learning is used to implement BIRCH, Adaptive random forest algorithm. Association mining is implemented using MLxTEND machine learning library. In the future we proposed recommendation model will be implemented as web service and will be tested with online web and mobile-based platform.

4. CONCLUSIONS

We have presented initial steps for building Garment fashion recommendation system by considering biometric profiles to segregate the data. Also customer's style preference corresponding to different fitting style has known by generating association rules. Proposed recommendation system has combined unsupervised, supervised and association learning to build the GFRS. GFRS system is developed to track user preferences corresponding to their selection behaviour on design elements corresponding to body measurement profile. Because customized garment online platforms are very specific to body type. BIRCH, Adaptive random forest classifier is used to make scaled system and to provide nearly real time prediction. In our future research work we will work with other customized garments to make this system more generalized. Recommendation service will be developed and will be tested with the online platform which connects different fashion brands(Project partners) and validates their business cases.

ACKNOWLEDGMENTS

We thank the European Union for providing an opportunity to contribute FBD BMODEL H2020 project

REFERENCES

- [1] Achim Berg, Imran Amed, Anita Balchandani, Johanna Andersson, Saskia Hedrich, and Robb Young. (2019) Fashion industry trends to watch in 2019 |McKinsey. [Online]. Available: <https://www.mckinsey.com/industries/retail/ourinsights/ten-trends-for-the-fashion-industry-to-watch-in-2019>
- [2] N. Wixcey. (2015) Made to order: The rise of mass personalisation | deloitte switzerland | consumer business. [Online]. Available:<https://www2.deloitte.com/ch/en/pages/consumer-business/articles/madeto-order-the-rise-of-mass-personalisation.html>
- [3] B. Touchette, Morgan Schanski, and Seung-Eun Lee, "Apparel brands's use of facebook: an exploratory content analysis of branded entertainment | emerald insight," vol. Vol. 19 No. 2, pp. 107-119. [Online]. Available: <https://www.emerald.com/insight/content/doi/10.1108/JFMM-04-2013-0051/full/html>
- [4] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," IEEE Internet computing, no. 1, pp. 76–80, 2003.
- [5] X. Chen, H. Chen, H. Xu, Y. Zhang, Y. Cao, Z. Qin, and H. Zha, "Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation," in Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR'19. New York, NY, USA: ACM, 2019, pp.765–774. [Online]. Available: <http://doi.acm.org/10.1145/3331184.3331254>

- [6] Ã. Cardoso, F. Daolio, and S. Vargas, "Product characterisation towards ;: Learning attributes from unstructured data to recommend fashion products," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18. ACM Press, pp. 80–89. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3219819.3219888>
- [7] Stacia Carr. Online clothing retailers hunt for better fit to cut costly returns - reuters. [Online]. Available: <https://www.reuters.com/article/us-onlineapparel-returns-focus/online-clothing-retailers-hunt-for-better-fit-to-cut-costly-returns-idUSKCN1OK1E2>
- [8] A. Kim, "Method of ordering a custom-made suit online," Feb. 28 2019, uS Patent App. 15/687,690.
- [9] H. Hwangbo, Y. S. Kim, and K. J. Cha, "Recommendation system development for fashion retail e-commerce," *Electronic Commerce Research and Applications*, vol. 28, pp. 94–101, 2018.
- [10] I. Amed, Johanna, ersson, A. Berg, M. Drageset, S. Hedrich, and S. Kappelmark. The state of fashion 2018: Renewed optimism for the fashion industry | McKinsey. [Online]. Available: <https://www.mckinsey.com/industries/retail/ourinsights/renewed-optimism-for-the-fashion-industry>
- [11] A. Bhattacharya, N. Chowdhury, and R. K De, "Comparative analysis of clustering and biclustering algorithms for grouping of genes: co-function and co-regulation," *Current Bioinformatics*, vol. 7, no. 1, pp. 63–76, 2012.
- [12] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *SIGMOD Rec.*, vol. 27, no. 2, pp. 94–105, Jun. 1998. [Online]. Available: <http://doi.acm.org/10.1145/276305.276314>
- [13] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," vol. 2, no. 2, pp. 165–193. [Online]. Available: <https://doi.org/10.1007/s40745-015-0040-1>
- [14] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," *SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996. [Online]. Available: <http://doi.acm.org/10.1145/235968.233324>
- [15] B. Karmakar and I. Mukhopadhyay, "An efficient partition-repetition approach in clustering of big data," in *Big Data Analytics*. Springer, 2016, pp. 75–93.
- [16] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," vol. 20, pp. 53 – 65. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0377042787901257>
- [17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [18] L. Breiman, "Bagging predictors," vol. 24, no. 2, pp. 123–140. [Online]. Available: <https://doi.org/10.1007/BF00058655> [19] ———, "Random forests," vol. 45, no. 1, pp. 5–32. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [20] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," vol. 106, no. 9, pp. 1469–1495. [Online]. Available: <https://doi.org/10.1007/s10994-017-5642-8>

- [21] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective personalization based on association rule discovery from web usage data," in Proceedings of the 3rd International Workshop on Web Information and Data Management, ser. WIDM '01. ACM, pp. 9–15, event-place: Atlanta, Georgia, USA. [Online]. Available: <http://doi.acm.org/10.1145/502932.502935>
- [22] S. Bayati, A. F. Nejad, S. Kharazmi, and A. Bahreininejad, "Using association rule mining to improve semantic web services composition performance," in Control and Communication 2009 2nd International Conference on Computer, pp. 1–5.
- [23] R. Agrawal and J. C. Shafer, "Parallel mining of association rules," IEEE Transactions on Knowledge & Data Engineering, no. 6, pp. 962–969, 1996.
- [24] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '00. ACM, pp. 1–12, event-place: Dallas, Texas, USA. [Online]. Available: <http://doi.acm.org/10.1145/342009.335372>
- [25] M. D. Buhmann et al., 'Recommender Systems', in Encyclopedia of Machine Learning, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2011, pp. 829–838.

AUTHORS

Shukla Sharma, Ph.D. Candidate, GEMTEX Laboratory, France



A NOVEL MACHINE LEARNING SYSTEM FOR SENTIMENT ANALYSIS AND EXTRACTION

Osama Mohammad Rababah¹, and Nour Alokaily²

¹Information Technology Department, The University of Jordan, Amman, Jordan

²School of archaeology and Tourism, The University of Jordan, Amman, Jordan

ABSTRACT

The huge volume of online reviews makes it difficult for a human to process and extract all significant information to make decisions. As a result, there has been a trend to develop systems that can automatically summarize opinions from a set of reviews. In this respect, the automatic classification and information extraction from users' comments, also known as sentiment analysis (SA) becomes vital to offer users the best responses to users' queries, based on their preferences. In this paper, a novel system that offers personalized user experiences and solves the semantic-pragmatic gap was presented. Having a system for forecasting sentiments might allow us, to extract opinions from the internet and predict online user's favorites, which could determine valuable for commercial or marketing research. The data used belongs to the tagged corpus positive and negative processed movie reviews introduced by Pang and Lee[1]. The results show that even when a small sample is used, sentiment analysis can be done with high accuracy if appropriate natural language processing algorithms applied.

KEYWORDS

Machine Learning, Big Data, Natural Language Processing, Sentiment Analysis

1. INTRODUCTION

Sentiment analysis consists of the usage of language processing, text analysis, and computational linguistics to identify subjective opinion. Usually, the new data entries are compared to already classified samples, which belong to the same category. SA is the procedure of determining the polarity or intention of a written text [2].

According to [3], SA includes five steps to analyze sentiment data. The first step begins with data collection which consists of collecting data from user-generated content contained in blogs, forums, and social media networks. The collected data can be messy and expressed by different methods or by using different words, slangs, and context of writing. Manual analysis of such an enormous amount of data is not possible and exhausting. As a result, text analytics and natural language processing are used to mine and classify the data. Secondly, is the text preparation step, that is consists of cleaning the extracted data before analysis. Non-textual contents and contents that are unsuitable for the study are documented and detached. The third step is emotion detection, in which the extracted sentences of the reviews and opinions are scrutinized; sentences with individual expressions are retained, and sentences with objective communication are

Natarajan Meghanathan et al. (Eds) : CSEIT, CMLA, NeTCOM, CIoT, SPM, NCS, WiMoNe, Graph-hoc - 2019 pp. 387-393, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91330

discarded. The fourth step is sentiment classification where personal sentences are classified in positive, negative, good, bad, like, dislike, but classification can be made by using multiple points. Finally, it is the presentation of the output step where the key objective of sentiment analysis here is to transform unstructured text into meaningful information. At the end of the study, the test results are displayed on graphs. Also, time can be analyzed and can be graphically displayed a sentiment timeline with the chosen values of frequency, percentages, and averages over time [2].

The efficient auto-summarization of texts is a separate field of study in the computational linguistics community. One of its main goals is to offer users a way to access the content of their interest in a quicker and more efficient way [3]. SA, on the other hand, aims to be able to divide correctly text data into categories based on the opinions the authors expressed about particular issues, using natural language. To be able to offer personalized user experiences, these two fields can be analyzed holistically [4]. The novel system proposed in this article does that by merging an auto-summarization algorithm with a sentiment analysis algorithm and examining the results using the relevant metrics.

Accessing and searching reviews is frustrating when users have an imprecise idea of the product or its features and they need a recommendation or a close match. Keyword-based search does not usually provide good results, as the same keywords can appear in both good and bad reviews[5]. Another challenge in understanding studies is that a reviewer's general rating might be dedicated to the product features in which might not be of interest to the user searching. Additional challenges include having the sentiment word with an opposite meaning in a particular domain. Sarcastic sentences may violate the meaning of sentences; therefore, close attention to the words used in such sentences is needed. Other issues include when people write a word in different means which may not give us an indication that it is the same word. People's methods of expression can be inconsistent while most of the traditional text processing methods hang on the fact that a minor variance between two pieces of text doesn't alter the meaning.

This paper is prearranged as follows. Section 2 is a literature review of relevant work. Section 3 gives an overview of the methodology we adopt for this research article. Sections 4 and 5 present test cases and results obtained by running the novel system. Finally, Section 6 demonstrates the conclusion.

2. STATE OF ART

The term SA first appeared in [5]. However, the research on sentiments appeared earlier [6]-[10]. The literature on SA focused on diverse fields, from computer science to management sciences, social sciences and business due to its importance to various tasks such as subjective expressions [11], sentiments of words [12], subjective sentences [13], and topics [5] [13] and [14].

SA can be approached in different manners, either by categorizing data into two groups: positive or negative[15] or by using numerous intermediary classes, such as the multiple stars reviews [1]. The sentiment classification approaches can be classified into machine learning, lexicon-based and hybrid approach [16].

The increase in new categories of online information also changes the type of summarization that is of interest. Summarization has newly been combined with work on SA [17]-[19]. Given the

numerous different reviews that one can find on the web, the problem is to identify common opinions. Some of the approaches that have been tried so far include: determining semantic properties of an object, defining the intensity of an opinion, and determining whether opinion is important. In this paper, we present a novel system that was uniting an auto-summarization algorithm with a SA algorithm to increase personalized user experience.

The auto-summarization of texts was done using the tools offered by the NLTK toolkit (NLTK.org) [20], which provide the opportunity to tag sentences syntactically and calculate word frequencies and perform stop word elimination, by using the pre-defined English corpora.

3. A NOVEL SYSTEM ARCHITECTURE

The figure below shows the system flowchart. The auto summarization consists of 6 steps which start from the original text document that is given as an argument (step 1) and generate a summary of that text by selecting the n most relevant sentences (where n is a user-defined variable) (step 6). Steps (2) to (5) encompass of sentence tagging and word frequency and relevance calculations.

The tagged texts are handled and then handed to a naïve Bayesian classifier, along with their tags as training data. Once the classifier has been trained, new observations are given for classification. Figure 1 shows the steps of auto-summarization and Sentiment analysis.

The measures used to scale the performance of the sentiment analysis were as follow:

The sensitivity or the true positive rate and sometimes called recall, measures the proportion of positives that are correctly identified. Recall is calculated as below:

$$\text{True Positives} / \text{TruePositives} + \text{FalseNegatives}.$$

The precision or the positive predictive value is the fraction of relevant retrieved instances such as the percentage of negative restaurant or movie reviews that are truly negative. Precision is calculated as below:

$$\text{True Negatives} / \text{TruePositives} + \text{FalsePositives}.$$

Accuracy is defined as the closeness of agreement between the result of a measurement and a true value [21]. Accuracy is calculated as below:

$$\text{True Positives} + \text{True Negatives} / \text{TruePositives} + \text{FalseNegatives} +$$

$$\text{TrueNegatives} + \text{FalsePositives}$$

True positives: positive comments correctly identified as positive.

True negatives: negative comments correctly identified as negative.

False positives: negative comments incorrectly identified as positive.

False negatives: positive comments incorrectly identified as negative.

The values that were found for each of these meters are shown and discussed in the results section of this paper.

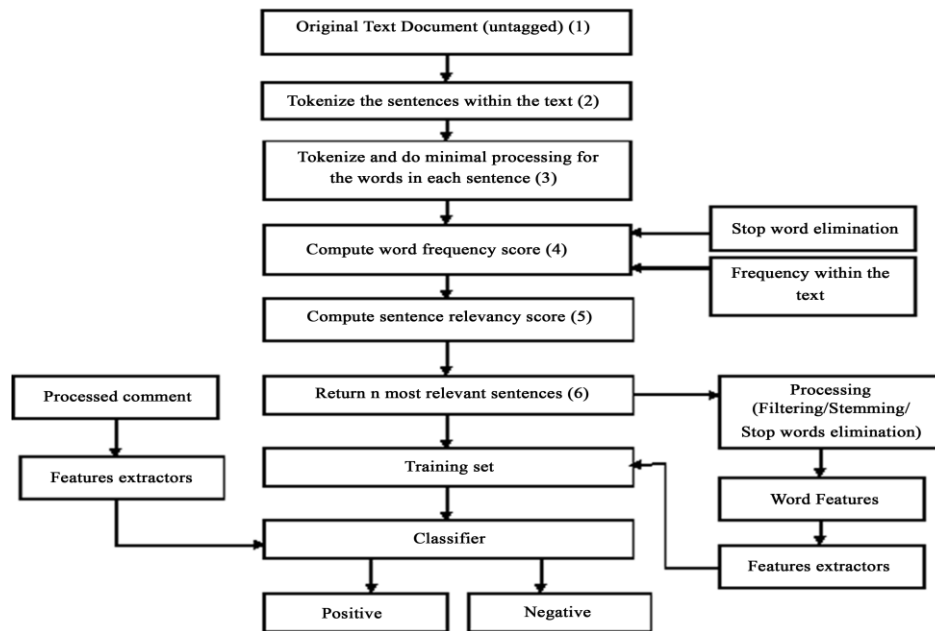


Figure 1. System flowchart.

4. SYSTEM TESTING

Five different forms of test cases were used:

- **No Proc process** which uses the original texts of the comments for both training and classification, with the dataset divided 20%/80%.
- **Min Proc process** only eliminates punctuation and uppercase letters, still uses the original complete textual comments for classification.
- **Sum on Sum** where all comments are summarized first and then they are used for training the Bayesian network and testing (again the 80%/20% ratio was used for the classification/testing).
- **Sum on full** where the Bayesian network is trained with the full text of the comments and the summaries are given as new items to be classified.
- **Full on Sum** where the Bayesian network is trained with the text of the summaries and the full textual comments texts are used for classification.

The 20%/80% ratio for training vs. classification was respected for all test cases, and no text was used for both training and testing (a summary of a text is considered the equivalent of the original text in this regard).

5. RESULTS

The results found from running the system for each of the test cases shown below.

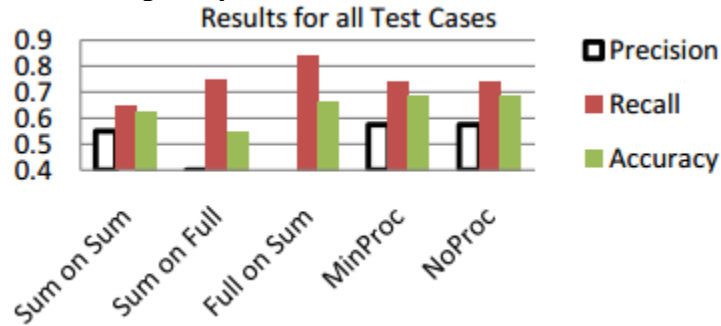


Figure 2. Complete value set for all test cases.

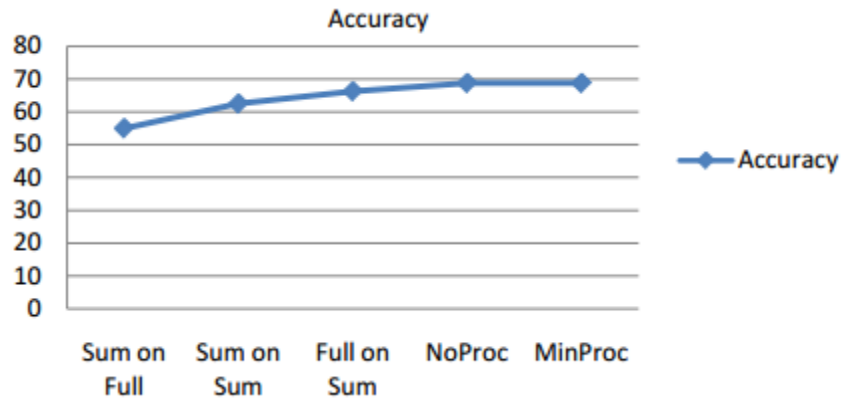


Figure 3. Accuracy depending on the test case

Table 1. Numeric values for all the test cases.

	Precision	Recall	Accuracy
Sum on Full	0.15	0.75	0.55
Sum on Sum	0.55	0.65	0.63
Full on Sum	0.4	0.84	0.66
NoProc	0.58	0.74	0.69

As shown in Table 1 and Figure 2 the best metrics were the Sum on Sum, Min- Proc, and NoProc, the top accuracy were NoProc and MinProc. Also, it's clear from Figure 3, the best accuracy was the NoProc and MinProc. Moreover, in contrast with the other metrics, accuracy has the minimum variation.

6. CONCLUSIONS

After running the system, results shows accuracy is improved when texts of the same type are used for training and testing. However, the accuracy of the system does not vary greatly between test cases—as opposed to the other metrics. Furthermore, there was no change in the results found for texts that were not processed and the ones that had undergone minimal processing; showing polarity were not influenced by the usage of upper case vs. lower case or punctuation signs.

The precision drops radically when the system is trained with different types of texts than the trained with (Sum on Full and Full on Sum test cases), the explanation being the same as the one for the accuracy drop. As further developments for the proposed system, the following directions could be investigated: 1) Variation of the number of sentences in the summaries depending on the length of the original text—assuring that the length of the original text does not affect the training algorithm; 2) Extra processing methods could be added to the algorithm, such as stemming and stop word elimination, and the results should be reexamined to determine if the performance metrics improve for the mix test cases—Sum on Full and Full on Sum.

ACKNOWLEDGEMENTS

This research is supported by The University of Jordan, Amman – Jordan.

REFERENCES

- [1] Pang, B. and Lee, L. (2004) A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. <http://dx.doi.org/10.3115/1218955.1218990>
- [2] Gupta, C., Jain, A., & Joshi, N. (2019). A Novel Approach to feature hierarchy in Aspect Based Sentiment Analysis using OWA operator. In Proceedings of 2nd International Conference on Communication, Computing and Networking (pp. 661-667). Springer, Singapore.
- [3] D'Andrea, A., Ferri, F., Grifoni, P. and Guzzo. T. (2015) Approaches, Tools and Applications for Sentiment Analysis Implementation. International Journal of Computer Applications, 125,26-33.
- [4] Abdulla, N., Ahmed, N., Shehab, M., AlAyyoub, M., Al-Kabi, M. and Al-Rifai, S. (2014) Towards Improving the Lexicon-Based Approach for Arabic Sentiment Analysis. International Journal of Information Technology and Web Engineering (IJITWE), 9, 55-71. <http://dx.doi.org/10.4018/ijitwe.2014070104>
- [5] Sindhu, R., Jamail, R. and Kumar, R. (2014) A Novel Approach for Sentiment Analysis and Opinion Mining. International Journal of Emerging Technology and Advanced Engineering, 4,522-527.
- [6] Nasukawa, T. and Yi, J. (2003) Sentiment Analysis: Capturing Favorability Using Natural Language Processing. Proceedings of the 2nd International Conference on Knowledge Capture, Florida, 23-25 October 2003, 70-77. <http://dx.doi.org/10.1145/945645.945658>
- [7] Morinaga, S., Yamanishi, K., Tateishi, K. and Fukushima, T. (2002) Mining Product Reputations on the Web. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 341-349. <http://dx.doi.org/10.1145/775047.775098>

- [8] Pang, B., Lee, L. and Vaithyanathan, S. (2002) Thumbs up? Sentiment Classification Using Machine Learning Techniques. Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing, 79-86.
- [9] Alaei, A. R., Becken, S., & Stantic, B. (2019). Sentiment analysis in tourism: capitalizing on big data. *Journal of Travel Research*, 58(2), 175-191.
- [10] Turney, P. (2002) Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the 40th ACL, 417-424.
- [11] Wiebe, J. (2000) Learning Subjective Adjectives from Corpora. Proceedings of National Conference on Artificial Intelligence.
- [12] Wilson, T., Wiebe, J. and Hoffmann, P. (2009) Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics*, 35, 399-433. <http://dx.doi.org/10.1162/coli.08-012-R1-06-90>
- [13] Peacock, D. C., & Khan, H. U. (2019). Effectiveness of Social Media Sentiment Analysis Tools with the Support of Emoticon/Emoji. In 16th International Conference on Information Technology-New Generations (ITNG 2019) (pp. 491-494). Springer, Cham.
- [14] Yi, J., Nasukawa, T., Niblack, W. and Bunescu, R. (2003) Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), Florida, 19-22 November 2003, 427-434. <http://dx.doi.org/10.1109/icdm.2003.1250949>
- [15] Hiroshi, K., Tetsuya, N. and Hideo, W. (2004) Deeper Sentiment Analysis Using Machine Translation Technology. Proceedings of the 20th International Conference on Computational Linguistics (COWLING 2004), Geneva, 23-27 August 2004, 494-500. <http://dx.doi.org/10.3115/1220355.1220426>
- [16] Govindarajan, M. (2013) Sentiment Analysis of Movie Reviews Using Hybrid Method of Naive Bayes and Genetic Algorithm. *International Journal of Advanced Computer Research*, 3, 139-145.
- [17] Albanese, M. (2013) A Multimedia Recommender System. *ACM Transactions on the Internet Technology (TOIT)*, 13, 1-32. <http://dx.doi.org/10.1145/2532640>
- [18] Carenini, G. and Cheung, J. (2008) Extractive vs. NLG-Based Abstractive Summarization of Evaluative Text: The Effect of Corpus Controversiality. Proceedings of the International Natural Language Generation Conference, 33-41. <http://dx.doi.org/10.3115/1708322.1708330>
- [19] Lerman, K., Blair-Goldensohn, S. and McDonald, R. (2009) Sentiment Summarization: Evaluating and Learning User Preferences. Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, 514-522. <http://dx.doi.org/10.3115/1609067.1609124>
- [20] NLTK.org. (n.d.). Retrieved 01 28, 2016. <http://www.nltk.org/index.html>
- [21] Jagdale, R. S., Shirsat, V. S., & Deshmukh, S. N. (2019). Sentiment analysis on product reviews using machine learning techniques. In *Cognitive Informatics and Soft Computing* (pp. 639-647). Springer, Singapore.

AUTHOR INDEX

<i>Abdelouahab Amira</i>	57
<i>Ahmed Saidi</i>	57
<i>Alex Liang</i>	203
<i>Aliya Tabassum</i>	249
<i>Andrei Petrescu</i>	73
<i>Ariel Jiang</i>	183
<i>Arto Toppinen</i>	233
<i>Bakkar LASKAR</i>	301
<i>Chris Blondia</i>	209
<i>Cristiana Carvalho</i>	173
<i>Darshan Adiga</i>	153
<i>Dimitri KONSTANTAS</i>	19
<i>Dirk Bähre</i>	189
<i>Dogukan Aksu</i>	313
<i>Durmus Harman</i>	189
<i>Erdal ÖZDOĞAN</i>	267
<i>Erik van der Sluis</i>	353
<i>Fangyan Zhang</i>	83,203
<i>Filipe Cabral Pinto</i>	173
<i>Gonçalo Machado</i>	173
<i>Guillaume de Moffarts</i>	91
<i>Guillaume Terrasson</i>	331
<i>Günther Schuh</i>	189
<i>Guo Yu</i>	45
<i>Haji Akhundov</i>	353
<i>Hamid Khemissa</i>	33
<i>Hao Yuan</i>	45
<i>Harutyun Berberyan</i>	117
<i>Hengliang Tan</i>	01
<i>Ilídio Oliveira</i>	173
<i>Isabel Borges</i>	173
<i>Jatin Chaudhary</i>	233
<i>Jeremy Van den Eynde</i>	209
<i>Jiao Du</i>	01
<i>Jieneng Chen</i>	45
<i>Jörg Heib</i>	189
<i>Jukka Heikkonen</i>	233
<i>Kimmo Myllymäki</i>	233
<i>Licia Sbattella</i>	101
<i>Ludovic Koehl</i>	373
<i>Luis Manuel Moreno Chacon</i>	331
<i>Matthias Möller</i>	189
<i>Merouane BOUZID</i>	301
<i>Michel Bakni</i>	331
<i>Mihai Carabas</i>	73

<i>Mottaqiallah Taouil</i>	353
<i>Mourad Oussala</i>	33
<i>Moussa WITTI</i>	19
<i>Muhammed Ali Aydin</i>	313
<i>Muzaffar Shah</i>	153
<i>Nour Alokaily</i>	387
<i>O.Ayhan ERDEM</i>	267
<i>Octavian Curea</i>	331
<i>Omar Nouali</i>	57
<i>Osama Mohammad Rababah</i>	387
<i>Pascal Bruniaux</i>	373
<i>Paul Scholz</i>	189
<i>Peng Zou</i>	139
<i>Qi Lu</i>	83, 183
<i>Rabah Attia</i>	221
<i>Rafflesia Khan</i>	287
<i>Raja Alaya</i>	221
<i>Rajeev Kanth</i>	233
<i>Rameswar Debnath</i>	287
<i>Ran Wei</i>	01
<i>Rene V.Mayorga</i>	167
<i>Roberto Tedesco</i>	101
<i>Said Hamdioui</i>	353
<i>Saroja Kanchi</i>	321
<i>Sebastian Schorr</i>	189
<i>Sébastien Combéfis</i>	91
<i>Sertap Kamçı</i>	313
<i>Shabir Bhat</i>	153
<i>Shahid Ali</i>	117
<i>Shubham Sharma</i>	167
<i>Shukla Sharma</i>	373
<i>Shuo Yang</i>	01
<i>Tuomas Korpi</i>	233
<i>Vincenzo Scotti</i>	101
<i>Viveka Vyeth</i>	153
<i>Wadha Lebda</i>	249
<i>Xianyi Zeng</i>	373
<i>Xiongda Chen</i>	45
<i>Yifan Ma</i>	45
<i>Yixuan Qi</i>	83
<i>You Peng</i>	183
<i>Yu Su</i>	83, 203
<i>Yudith Cardinale</i>	331
<i>Yunfei Cai</i>	139