Computer Science & Information Technology 85

David C. Wyld
Jan Zizka (Eds)

Computer Science & Information Technology

4th International Conference on Image Processing and Pattern Recognition
(IPPR 2018) April 28~29, 2018, Copenhagen, Denmark.

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Jan Zizka,
Mendel University in Brno, Czech Republic
E-mail: zizka.jan@gmail.com

ISSN: 2231 - 5403
ISBN: 978-1-921987-83-0
DOI : 10.5121/csit.2018.80501 - 10.5121/csit.2018.80514

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

Preface

The 4th International Conference on Image Processing and Pattern Recognition (IPPR 2018) was held in Copenhagen, Denmark, during April 28~29, 2018. The 4th International Conference on Software Engineering (SOENG 2018), The 4th International Conference on Data Mining (DaMi 2018), The 5th International Conference on Computer Science and Information Technology (CSIT 2018), The 4th International Conference on Artificial Intelligence and Soft Computing (AIS 2018), The 6th International Conference on Computational Science and Engineering (CSE 2018), The 5th International Conference on Signal Processing (CSIP 2018) and The 5th International Conference on Computer Networks & Communications (CCNET 2018) was collocated with The 4th International Conference on Image Processing and Pattern Recognition (IPPR 2018). 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 IPPR-2018, SOENG-2018, DaMi-2018, CSIT-2018, AIS-2018, CSE-2018, CCNET-2018 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. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, CCSEA-2018, NCOM-2018, AIFU-2018, DKMP-2018, EMSA-2018, SIPRO-2018, SEA-2018 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 CCSEA-2018, NCOM-2018, AIFU-2018, DKMP-2018, EMSA-2018, SIPRO-2018, SEA-2018.

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.

David C. Wyld
Jan Zizka

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Abhishake	Indian Institute of Technology - Delhi, India
Ahmad Qawasmeh	The Hashemite University, Jordan
Alex Afanasyev	Florida International University, U.S.A
Amizah Malip	University of Malaya, Malaysia
Asad Abdi	The University of Technology, Malaysia
Ayush Singhal	University of Minnesota, Minnesota
Basar Oztaysi	Istanbul Technical University, Turkey
Biju Issac	Teesside University, UK
Bing Zhou	Sam Houston State University, USA
Bouchra Marzak	Faculty of Sciences - Hassan II University, Morocco
Carmen Martinez	University of Jaen, Spain
Christophe NICOLLE	University of Bourgogne, France
Da Yan	The University of Alabama at Birmingham, USA
Dabin Ding	University of Central Missouri, USA
Dariusz Jacek Jakobczak	Koszalin University of Technology, Poland
Deepak Laxmi Narasimha	University of Malaya
Denivaldo Lopes	Federal University of Maranhao, Brazil
Edwin Lughofer	Johannes Kepler University Linz, Austria
Erman Çakit	Aksaray University, Turkey
Farhad pourfarzi	Ardabil University of Medical Sciences, Iran
Figen Balo	Firat University, Turkey
Findler	University of Houston Clear Lake, USA
Gazi Erkan Bostanc	Ankara University, Turkey
Grigorios N. Beligiannis	University of Patras, Greece
Guoqing Xiao	Hunan University, China
Hadi Amirpour	Universidade da Beira Interior, Portugal
Haibo Yi	Shenzhen Polytechnic, China
Haipeng Cai	Washington State University, USA
Hamid Ali Abed AL-Asadi	Basra University, Iraq
Hamid Rastegari	Islamic Azad University, Iran
Harish Garg	Thapar Institute of Engineering and Technology, India
Hector Migallon	Miguel Hernandez University, Spain
Hiromi Ban	Nagaoka University of Technology,Japan
Hossein Ghaffariang	Arak University, Iran
Houda KHROUF	Atos Innovation Lab, France
I-Ching Hsu	National Formosa University, Taiwan
I-Hsien Ting	National University of Kaohsiung, Taiwan

Irena Patasiene	Kaunas University of Technology, Lithuania
Isa Maleki	Islamic Azad University, Iran
Israa Sh. Tawfic	University of Gaziantep,Turkey
Issa Atoum	The World Islamic Sciences and Islamic Studies, Jordan
Iyad Alazzam	Yarmouk University,Jordan
Janes Andrea	Free University of Bozen-Bolzano, Italy
Jasmine Seng K. P.,	Charles Sturt University, Australia
Jatindra Kumar Deka	Indian Institute of Technology Guwahati, India
Jia Zhu	South China Normal University, China
Jonice Oliveira	Universidade Federal do Rio de Janeiro (UFRJ), Brazil
Joshua (Songqing) Yue	University of Wisconsin-Platteville, USA
Jun Liu	University of Michigan at Dearborn, USA
Karina Gibert	Universitat politecnica de catalunya, Spain
Katarzyna Rudnik	Opole University of Technology, Poland
Kevin Gary,	Arizona State University, USA
Khader Mohammad	Birzeit University, Palestine
Manuel Angel Serrano Martin	Universidad de Castilla, Spain
Marius CIOCA	Lucian Blaga University of Sibiu, Romania
Mark Robert Anderson	Edge Hill University, UK
Maryam Habibi	Humboldt-Universitat zu Berlin, Germany
Mohammad Hamdan	Heriot Watt University, UAE
Mohammadreza Balouchestani	Indiana Purdue Fort Wayne University, USA
Mohammed Al-Mai'itah	Al-Balqa applied university, Jordan
Murat PEKER	Niğde Omer Halisdemir University, Turkey
Nadhir Ben Halima	Taibah University, Saudi Arabia
Nahlah Shatnawi	Yarmouk University, Jordan
Nasser Thabet	Szabist University, UAE
Nicolas H. Younan	Mississippi State University, USA
Nizar Chatti	University of ANGERS, France
Oded Maimon	Tel Aviv university, Israel
Osama Rababah	The University of Jordan, Jordan
Panos M. Pardalos	University of Florida,USA
Peiman Mohammadi	Islamic Azad University, Iran
Pr. Smain FEMMAM	UHA University ,France
Rafa E.Al-Qutaish	ETS - University of Quebec, Canada
Rajarajan M	City University, United Kingdom
Reza Ebrahimi Atani	University of Guilan, Iran
Rhattoy A	Higher School of Technology, Morocco
Ruhaya Ab. Aziz	University Tun Hussein Onn Malaysia
Saed TARAPIAH	An-najah National University, Palestine
Sanjay Kumar Biswas	San Diego State University, USA
Shah J Miah	Victoria University, Australia
William Chueh	Stanford University, USA
William R. Simpson	Institute for Defense Analyses, USA
Xingfu Wang	University of Science and Technology of China, china
Yashar Deldjoo	Universita degli Studi di Milano-Bicocca Milan, Italy
Zeshui Xu	Sichuan University, China
Zeyu Sun	Luoyang Institute of Science and Technology, China
Zhao Liang	University of Sao Paulo,Brazil

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

4th International Conference on Image Processing and Pattern Recognition (IPPR 2018)

4th International Conference on Software Engineering (SOENG 2018)

4th International Conference on Data Mining (DaMi 2018)

5th International Conference on Computer Science and Information Technology (CSIT 2018)

4th International Conference on Artificial Intelligence and Soft Computing (AIS 2018)

6th International Conference on Computational Science and Engineering (CSE 2018)

5th International Conference on Signal Processing (CSIP 2018)

5th International Conference on Computer Networks & Communications (CCNET 2018)

EMPIRICAL COMPARISON OF VISUAL DESCRIPTORS FOR ULCER RECOGNITION IN WIRELESS CAPSULE ENDOSCOPY VIDEO

Ouiem Bchir[1], Mohamed Maher Ben Ismail[1] and Nourah AL_Aseem[1,2]

[1]Computer Science Department, College of Computer and Information
Sciences, King Saud University
[1,2]Computer Science Department,
College of Engineering and Computer Sciences,
Prince Sattam Bin Abdulaziz University

ABSTRACT

In this work, we empirically compare the performance of various visual descriptors for ulcer detection using real Wireless Capsule Endoscopy WCE video frames. This comparison is intended to determine which visual descriptor represents better WCE frames, and yields more accurate gastrointestinal ulcer detection. The extracted visual descriptors are fed to the ulcer recognition system which relies on Support Vector Machine (SVM) classification to categorize WCE frames as "ulcer" or "non-ulcer".

KEYWORDS

Visual descriptors, Ulcer detection, Wireless Capsule Endoscopy.

1. INTRODUCTION

A disease can be defined as a particular abnormal case, a disorder of structure or function, which impacts a specific side or all of an organism [1].Wireless Capsule Endoscopy (WCE) is an advanced technology which is used to recognize internal diseases[2], especially ulcer of the digestive tract. Its main advantages are flexibility, accuracy, pain free, and reasonable cost [3]. WCE system consists of three components: an electronic capsule sensing system, a recording device, and a computer for image review and interpretation[4].More specifically, WCE is a small capsule containing a miniature camera that is swallowed by the patient, and captures around 50,000 frames of the gastro-intestinal track that are transmitted to the receiver in a real time manner. The video is then uploaded to a computer for examination by the physician in order to identify gastrointestinal diseases. However, the recorded video is too long which makes its review awkward and time-consuming for physicians. Thus, reducing the examination time is required to make the analysis of the video less tedious[4].

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 01–09, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80501

Various types of ulcers which may affect the digestive tract can be detected using WCE. Namely, these types include: Peptic [5], Gastric[6], Duodenal[7], and Esophageal[8]. A peptic ulcer is the degradation of a tissue area by gastric juices that are produced by the stomach and the intestines to digest food. When the immunity system is weak, gastric juices attack the envelope of the gastro-intestinal track and results on peptic ulcers [9]. It is defined as arupture in the mucosal lining of the stomach [3]. For the case of Gastric ulcer, the tear is in the stomach[6]. Duodenal ulcer appears at the beginning of the small intestine [7] as an opening in the duodenum. A complicated case of acid reflux may lead Esophageal ulcer that appears at the extreme end of the esophagus[8].

Low-level feature extraction is one of the main components of any image analysis system. Their role is to convey the visual properties of an image to the recognition phase. However, choosing the appropriate visual descriptor for a specific recognition problem remains a challenging task for pattern recognition researchers. In particular, for ulcer detection, low-level visual descriptors extracted from the WCE frames do not encode and convey the same visual information to the recognition system. Thus, they do not contribute equally in the recognition power of the system. The keystone is then to identify the most discriminating visual descriptors. In other words, the answer to the question “Which visual descriptor yields the most accurate ulcer detection in WCE video frames” should be answered[6, 10]. Despite the researchers' efforts[11-12]to propose specific visual descriptors able to recognize ulcer in WCE video, no objective answer has been given to the question “What is the best descriptor to detect ulcer in WCE video?”. Moreover, none of the existing research compares the discrimination performance of these visual descriptors. In this research a comparison between the visual descriptors, used for ulcer detection WCE video frames, is conducted in order to determine which one discriminates the best between ulcer and ulcer-free frames.

The rest of this report is organized as follows. In section 2 we outline existing ulcer recognition techniques using WCE data. The Empirical comparison of visual descriptors for ulcer recognition is reported and analyzed in section 3. Finally, section 3 concludes this work and outlines potential future works.

2. LITERATURE REVIEW

Detecting and recognizing digestive ulcer using digital images as data modality is an active research field which has been promoted to assist physicians. This support aims at detecting ulcer using digestive endoscopy images [13]. During the last decade, various digestive ulcer detection techniques have been developed, and Wireless Capsule Endoscopy (WCE) emerged as an effective diagnostic tool. This technique enables doctors to gather much more gastric images. After obtaining the images, they are pre-processed. Then, visual descriptor extraction techniques are applied to encode the visual properties of the images. Finally, a supervised learning technique is launched to automatically detect ulcer frames.

In[14, 15], the researchers proposed texture visual descriptors to distinguish ulcer regions from normal regions. They used wireless capsule to take images from the inside of stomach and save them in a database. Then, new images are compared with this database in order to detect the place of the ulcer. In[14], the texture visual descriptors used to describe the images patterns are Curvelet transform[16] and local binary pattern[17]. The researchers used Neural Network [18]to classify the extracted visual descriptors and determine if there is an ulcer or not. In[15], the researchers used Curvelet transform, local binary pattern and YCbCr color[17]along with Neural Network and Support Vector Machines (SVM) to classify the images and determine if there is an ulcer or not[18, 19]. The visual descriptor used in [15] yield better accuracy than those in[14].

The researchers in [33]intended to recognize the gastrointestinal tract (GI) ulcer using digital image processing techniques. They used LM-LBP filter bank and local binary pattern as texture visual descriptor to recognize ulcer area[21]. Also, they used Image Block Dictionary (IBD) classifier and K-means clustering algorithm for training[21]. They extracted a significant number of image blocks (of size 128 by 128 pixels) of abnormal and normal textures from WCE and colonoscopy images. Then, the extracted image blocks were verified using the domain experts. Also, they used fixed size image blocks instead of regions obtained using segmentation algorithm, because it is simple to implement and fast to compute. In the detection phase, they evaluated unseen image in order to estimate whether it is abnormal or normal. After that, the image is scanned row and column wise using image blocks having a predefined overlap with the previous scan. A set of images was extracted from five real WCE videos for each abnormal and normal textures.

In[22], the authors studied the detection of gastrointestinal tract (GI) ulcer using color pattern as low-level visual descriptor. They used CIE-lab color visual descriptor[23]to represent WCE frames. The researchers used a total of 1370 representative frames captured during 252 WCE procedures with MiroCam [24]at the Royal Infirmary of Edinburgh. They used the MiroCam capsule endoscope[24] which has a frame rate of 3 frames per second, and an image resolution of 320x320 pixels. Also, they used Support Vector Machine (SVM) to classify the WCE frames and discriminate between pathology and normal frames. Another gastrointestinal ulcer recognition is proposed in[25]. The researchers used texture and color visual descriptors along with an SVM model to classify ulcer frames. The researchers concluded that a combination of texture and color descriptors enhances ulcer detection in WCE video. The authors in[26]aimed to recognize bleeding ulcer. In their study, chromaticity moments [30] were extracted as color visual descriptor, and a Multi-layer perceptron (MLP) Neural Network[28] was used to classify WCE frames. In[29], the Color Coherence Vector (CCV) visual descriptor[30] was used to classify bleeding ulcer using SVM. The researchers considered 220 images of bleeding, 159 images of ulcers, and 228 images of non-bleeding/ulcers. Thiers images were captured in the small intestine by using the PillCam SB WCE[31]. Similarly, the authors in[11] worked on the recognition of peptic ulcer in WCE video. They used HSV color and texture visual descriptor as visual descriptors. Also, they used Support Vector Machines (SVMs) to classify the images the ulcer frames. The researchers managed to run their algorithm on 20 frames with ulcer cases. Besides that, they used 10 extra frames with ulcer cases to build the SVM model. The work in[32] focused on ulcer recognition using bag-of-words, LBP and SIFT [33]. The authors proposed a visual descriptor fusion technique to aggregate the different low-level visual descriptors, and represent the content of each frame using one single vector. Finally, they used Support Vector Machines (SVMs) to classify the frame instances. In [34], the researchers used MPEG-7 Visual Descriptors[35] to detect specific anomalies such as blood and ulcers. The result showed that the Scalable Color and Homogenous texture descriptors yield the better performance measures.

3. EMPIRICAL COMPARISON OF VISUAL DESCRIPTORS FOR ULCER RECOGNITION

We study several visual descriptors in order to find the most discriminating ones in terms of ulcer detection performance. First, we collect real WCE image to assess visual descriptors. The ground truth is provided for each frame. It consists of a label encoding if the frame contains an ulcer or not. In our experiments, we run a *k-fold* cross validation with *k*=10. We used three performance measures. Namely, sensitivity, accuracy and Specificity. These performance measures aim at evaluating the discriminating power of the visual descriptors with respect to ulcer and non-ulcer frames. In other words, the visual descriptors witch yields the best performance measures with be considered as the most discriminating ones and would be recommended for automatic ulcer detection in WCE frames.

3.1 Data Set

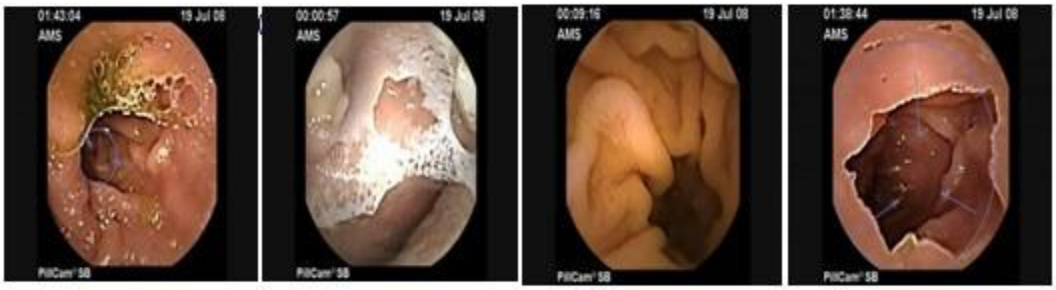


Figure 1.Sample ulcer images from WCE video.(a) Bleeding ulcer, (b)Esophageal ulcer, (c) Gastric ulcer, (d) Peptic ulcer.

We use a real WCE frames to run an experiments [36]. The dataset consists of 4274 images taken using WCE video [36], where 4024 are ulcer images, and 350 are normal. Figure 1 shows the sample ulcer images from the data set. These images taken from different parts of the digestive system and describes the appearance of different types of ulcer disease.

3.2 Experiment Description

We use various visual descriptors along with SVM classifier to find the most accurate one, and recommend it for detecting ulcer frames in WCE video. Namely, we consider Local Binary Pattern [37], Curvelet Transform [38], Chromaticity Moments Color [39], Color Coherence Vector [27], Homogenous Texture Descriptor [40], Scalable Color Descriptor [41], Ycbcr Color Histogram [42], the CIE_Lab Color Histogram [43], and the HSV Color Histogram [44]. The visual descriptors extracted from WCE frames are provided as an input to the SVM classifier, and the performance of the ulcer detection process is assessed with respect to each visual descriptor. When a descriptor yields low classification performance, it can be concluded that it was not able to convey the appropriate relevant information to the SVM classifier. Moreover, it can be claimed that the visual descriptor space of this visual descriptor does not represent efficiently "ulcer" and "non-ulcer" classes. Frame classification using SVM consists in determining the optimal hyperplane which separates data instances from both classes in the considered visual descriptor space. The optimal hyperplane corresponds to the widest margin between the two categories.

3.3 Experiments Results

The visual descriptors assessment results are summarized in Table 1. As one can notice in Table 1, the sensitivity level achieved by all visual descriptors is relatively high, and does not allow to objectively compare the obtained performances. On the other hand, the accuracy and the specificity show relevant variation, and reflect different performance levels of the different descriptors. In particular, the curvelet transform, the Homogeneous Texture Descriptor and the HSV color histograms achieved the poorest ulcer detection performance in terms of accuracy, and were drastically outperformed by the rest of the low-level visual descriptors. Also, these three descriptors along with the Chromaticity Moments Color, the Scalable Color Descriptor, the Color Coherence Vector, and the Ycbcr Color Histogram attain relatively low specificity levels. This means that these descriptors miss-classify a high portion of the "non-ulcer" frames. On the other hand, the Local Binary Pattern and the CIE_Lab Color Histogram are the two visual descriptors that were able to discriminate the best between "ulcer" and "non-ulcer" WCE video frames. In other words, they were able to represent the WCE video frames in their corresponding visual descriptor spaces in a away, they allow the SVM classifier to accurately separate between the two classes. Figure 2 shows the corresponding ROC curves.

Table 1.Accuracy, Sensitivity, and Specificity obtained for all visual descriptors using SVM classifier.

Visual descriptor	Accuracy	Sensitivity	Specificity
Local Binary Pattern	98.85%	99.4%	90.03%
Curvelet Transform	47.82%	99.22%	9.62%
Chromaticity Moments Color	77.42%	99.54%	13.79%
Color Coherence Vector	82.87%	99.93%	25.35%
Homogenous Texture Descriptor	48.50%	99.40%	09.83%
Scalable Color Descriptor	72.46%	99.79%	17.24%
Ycbcr Color Histogram	65.47%	99.96%	14.44%
CIE_Lab Color Histogram	98.95%	99.06%	96.80%
HSV Color Histogram	53.74%	96.50%	8.34%

As it can be seen, the results in Figure 2 confirms the results in Table 1. More specifically, the Local Binary Pattern and the CIE-Lab color moments outperform the other visual descriptors in terms of accuracy and sensitivity. This performance can be attributed to the fact that they are not sensitive to the monotonic color level variations that is caused by the high illumination variance of the WCE video frames that were captured at different locations of the gastrointestinal tract. Moreover, these results confirm that they visual descriptors satisfy the perceptual uniformity principle, and ensure that the difference between two patterns, as perceived by the human eye, is proportional to the distance measured within these two visual descriptor spaces.

CIE-Lab color descriptor outperforms the others colors descriptors because it is designed to model letter the human perception. Also, by definition in CIE-Lab, a color is either red or green, blue or yellow. This is appropriate to WCE frame characteristics where only red and yellow colors are present while no green or blue colors are included.

On the other hand, LBP texture descriptor beats the other texture descriptors because it encodes a combined structural and statistic of the texture pattern. Moreover, LBP provides a local information about the texture that can capture the visual properties of the WCE video frames containing ulcer symptoms.

While these two visual descriptors attain almost the same accuracy and sensitivity values, CIE-Lab color histogram achieves a higher specificity rate. Thus, it mis-classified less normal frames as “ulcer” than LBP based classification. Practically, this means that it would reduce the risk of further examination of healthy patient that were detected as “ulcer” cases.

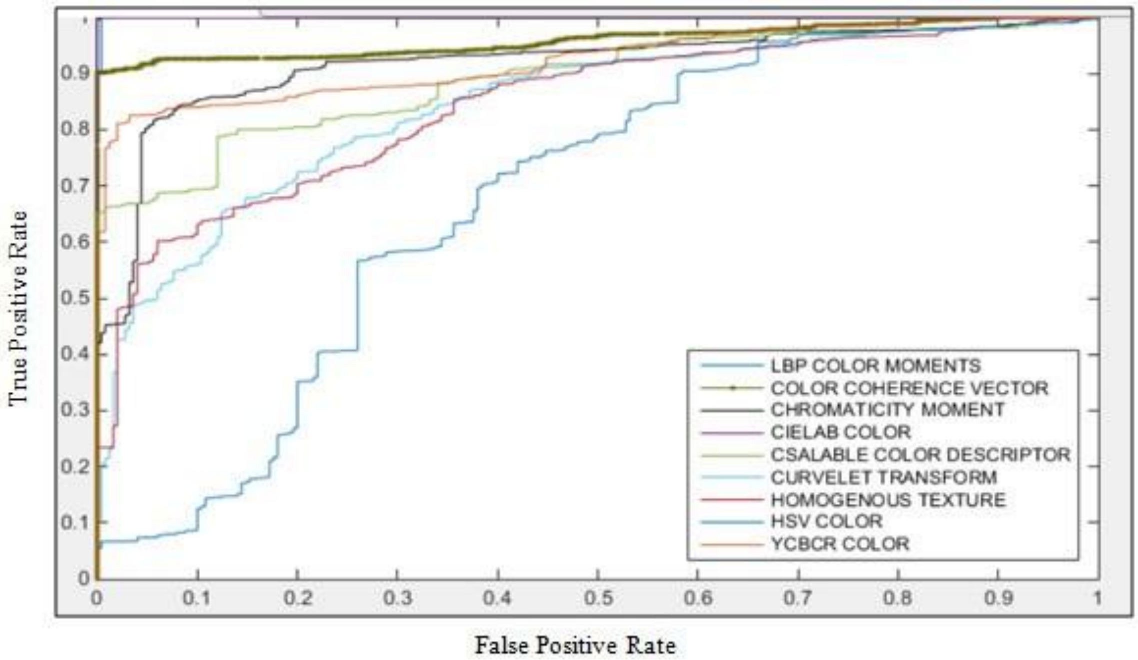


Figure 2. ROC curve obtained using the different visual descriptors and SVM classifier

4. CONCLUSIONS AND POTENTIAL FUTURE WORKS

Wireless Capsule Endoscopy (WCE) is the latest technology able to screen intestinal pathologies at an early stage. Despite its convenience to patients and its effectiveness to show small intestinal details, the physician involvement remains tedious and time consuming. The pattern recognition system can assist the physician by automatically detecting ulcer in WCE videos. However, the performance of such systems is sensitive to the choice of the visual descriptors. In fact, each feature encodes specific visual properties, and provides different discriminative information to the detection algorithm. The keystone is then to determine the feature(s) which yield(s) better recognition of the Ulcer pattern.

Our results showed that LBP and CIE-Lab color histogram outperform the other visual descriptors and achieved the best performance. The robustness of these two low-level features to monotonic color level variations caused by the high illumination variance in the gastrointestinal tract, along with their satisfaction of the perceptual uniformity principle, are the main properties that yield the obtained results.

Thus, we can proclaim that LBP and CIE-Lab color histogram are the most discriminating visual descriptors between “ulcer” and “non ulcer” frames in WCE video.

As potential future work, we plan to investigate visual descriptor aggregation in order to enhance the ulcer detection accuracy. This approach involves fusion techniques to optimize the combination of descriptors to best represent the visual content of the WCE video frames. For instance, efficient aggregation of LPB and CIE-Lab color histogram may enhance the overall ulcer detection accuracy.

REFERENCES

[1] A. S. Levey, K.-U. Eckardt, Y. Tsukamoto, A. Levin, J. Coresh, J. Rossert, D. d. Zeeuw, T. H. Hostetter, N. Lameire, and G. Eknoyan, (2005) "Definition and classification of chronic kidney disease: a position statement from Kidney Disease: Improving Global Outcomes (KDIGO)," Kidney international, vol. 67, pp. 2089-2398.

[2] C. Jingfeng, "Medicine in China," Encyclopaedia of the History of Science, Technology, and Medicine in Non-Western Cultures, pp. 1529-1534, 2008.

[3] G. Iddan, G. Meron, A. Glukhovsky, and P. Swain, (2000) "Wireless capsule endoscopy," Nature, vol. 405, pp. 417-417.

[4] G. D. Finlayson, S. D. Hordley, and I. Tastl, (2006) "Gamut constrained illuminant estimation," International Journal of Computer Vision, vol. 67, pp. 93-109.

[5] Z. Fireman, A. Glukhovsky, H. Jacob, A. Lavy, S. Lewkowicz, and E. Scapa, (2002) "Wireless capsule endoscopy," IMAJ-RAMAT GAN-, vol. 4, pp. 717-719.

[6] T. Gevers and A. W. Smeulders, "Color-based object recognition, (1999) " Pattern recognition, vol. 32, pp. 453-464.

[7] S. A. Shafer, "Using color to separate reflection components, (1985)" Color Research & Application, vol. 10, pp. 210-218.

[8] T. Ojala and M. Pietikäinen, (1999)"Unsupervised texture segmentation using feature distributions," Pattern recognition, vol. 32, pp. 477-486.

[9] peptic ulcer. Available: www.health.harvard.edu/digestive ../peptic-ulcer

[10] J. Oh, (2013) "Detection of temporal events and abnormal images for quality analysis in endoscopy videos," UNIVERSITY OF NORTH TEXAS.

[11] A. Karargyris and N. Bourbakis, (2009) "Identification of ulcers in wireless capsule endoscopy videos," in Biomedical Imaging: From Nano to Macro, pp. 554-557.

[12] J.-Y. Yeh, T.-H. Wu, and W.-J. Tsai, (2014) "Bleeding and ulcer detection using wireless capsule endoscopy images," Journal of Software Engineering and Applications, vol. 7, p. 422.

[13] B. Marshall, J. R. Warren, E. Blincow, M. Phillips, C. S. Goodwin, R. Murray, S. Blackbourn, T. Waters, and C. Sanderson, (1988) "Prospective double-blind trial of duodenal ulcer relapse after eradication of Campylobacter pylori," The Lancet, vol. 332, pp. 1437-1442.

[14] B. Li and M.-H. Meng, (2008) "Ulcer recognition in capsule endoscopy images by texture features, WCICA 2008, pp. 234-239.

[15] B. Li and M. Q.-H. Meng, (2009) "Texture analysis for ulcer detection in capsule endoscopy images," Image and Vision computing, vol. 27, pp. 1336-1342.

[16] E. J. Candes and D. L. Donoho, (2000) "Curvelets, multiresolution representation, and scaling laws," in Proc. SPIE, pp. 1-12.

[17] T. Ojala, M. Pietikäinen, and D. Harwood, (1996) "A comparative study of texture measures with classification based on featured distributions," Pattern recognition, vol. 29, pp. 51-59.

[18] P. Wilding, M. A. Morgan, A. E. Grygotis, M. A. Shoffner, and E. F. Rosato, (1994) "Application of backpropagation neural networks to diagnosis of breast and ovarian cancer," Cancer Letters, vol. 77, pp. 145-153.

[19] J. A. Suykens and J. Vandewalle, (1999) "Least squares support vector machine classifiers," Neural processing letters, vol. 9, pp. 293-300.

[20] R. Nawarathna, J. Oh, J. Muthukudage, W. Tavanapong, J. Wong, P. C. De Groen, and S. J. Tang, (2014) "Abnormal image detection in endoscopy videos using a filter bank and local binary patterns," Neurocomputing, vol. 144, pp. 70-91.

[21] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons, (2001) " International Journal of Computer Vision, vol. 43, pp. 29-44.

[22] D. K. Iakovidis and A. Koulaouzidis, (2014) "Automatic lesion detection in capsule endoscopy based on color saliency: closer to an essential adjunct for reviewing software," Gastrointestinal endoscopy, vol. 80, pp. 877-883.

[23] G. Wyzecki and W. Stiles, (1982) "Color science: concepts and methods, quantitative data and formulae," New York, London, Sidney.

[24] L. Korman, M. Delvaux, G. Gay, F. Hagenmuller, M. Keuchel, S. Friedman, M. Weinstein, M. Shetzline, D. Cave, and R. de Franchis, (2005) "Capsule endoscopy structured terminology (CEST): proposal of a standardized and structured terminology for reporting capsule endoscopy procedures," Endoscopy, vol. 37, pp. 951-959.

[25] P. Szczypiński, A. Klepaczko, M. Pazurek, and P. Daniel, (2014) "Texture and color based image segmentation and pathology detection in capsule endoscopy videos," Computer methods and programs in biomedicine, vol. 113, pp. 396-411.

[26] B. Li and M. Q.-H. Meng, (2009) "Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments," Computers in Biology and Medicine, vol. 39, pp. 141-147.

[27] J.-M. Geusebroek, R. Van den Boomgaard, A. W. M. Smeulders, and H. Geerts, (2001) "Color invariance," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, pp. 1338-1350.

[28] S. Haykin, (1996), "Neural Networks: A Comprehensive Foundation", second edition ed. NewJersey: Prentice-Hall.

[29] J.-Y. Yeh, T.-H. Wu, and W.-J. Tsai, (2014) "Bleeding and Ulcer Detection Using Wireless Capsule Endoscopy Images," Journal of Software Engineering and Applications, vol. 7, pp. 422-432.

[30] G. Pass, R. Zabih, and J. Miller, (1997) "Comparing images using color coherence vectors," in Proceedings of the fourth ACM international conference on Multimedia, pp. 65-73.

[31] A. Moglia, A. Menciassi, and P. Dario, (2008) "Recent patents on wireless capsule endoscopy," Recent Patents on Biomedical Engineering, vol. 1, pp. 24-33.

[32] L. Yu, P. C. Yuen, and J. Lai, (2012) "Ulcer detection in wireless capsule endoscopy images," ICPR 2012, pp. 45-48.

[33] T.-M. Tu, P. S. Huang, C.-L. Hung, and C.-P. Chang, (2004) "A fast intensity-hue-saturation fusion technique with spectral adjustment for IKONOS imagery," Geoscience and Remote Sensing Letters, IEEE, vol. 1, pp. 309-312.

[34] M. T. Coimbra and J. S. Cunha, (2006) "MPEG-7 visual descriptors—contributions for automated feature extraction in capsule endoscopy," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 16, pp. 628-637.

[35] S.-F. Chang, T. Sikora, and A. Purl, (2001) "Overview of the MPEG-7 standard," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 11, pp. 688-695.

[36] DR.Khuroos. (16/12/2015,4:30 PM). http://drkhuroo.in/index.php

[37] E. Candes, L. Demanet, D. Donoho, and L. Ying, (2006) "Fast discrete curvelet transforms," Multiscale Modeling & Simulation, vol. 5, pp. 861-899.

[38] M. Choi, R. Y. Kim, and M.-G. Kim, (2004) "The curvelet transform for image fusion," International Society for Photogrammetry and Remote Sensing, ISPRS 2004, vol. 35, pp. 59-64.

[39] Paschos, "Fast color texture recognition using chromaticity moments, (2000) " Pattern Recognition Letters, vol. 21, pp. 837-841, 2000.

[40] Y. M. Ro, M. Kim, H. K. Kang, B. Manjunath, and J. Kim, (2001) "MPEG-7 homogeneous texture descriptor," ETRI journal, vol. 23, pp. 41-51.

[41] B. S. Manjunath, P. Salembier, and T. Sikora,(2002) "Introduction to MPEG-7: multimedia content description interface", vol. 1: John Wiley & Sons.

[42] S. Sural, G. Qian, and S. Pramanik, (2002) "Segmentation and histogram generation using the HSV color space for image retrieval," in Image Processing.

[43] G. J. Braun, M. D. Fairchild, and F. Ebner, (1998) "Color gamut mapping in a hue-linearized CIELAB color space," in Color and Imaging Conference, pp. 163-168.

[44] K. Cantrell, M. Erenas, I. de Orbe-Paya, and L. Capitán-Vallvey, (2009) "Use of the hue parameter of the hue, saturation, value color space as a quantitative analytical parameter for bitonal optical sensors," Analytical chemistry, vol. 82, pp. 531-542.

INTENTIONAL BLANK

ITERATIVE HAAR-DWT BASED EFFICIENT IMAGE STEGANOGRAPHY

Aditi Singh[*1], K S Venkatesh[2] and Vikas Patidar[3]

Department of Electrical Engineering,
Indian Institute of Technology Kanpur, Kanpur-208016, Uttar Pradesh, India

ABSTRACT

In image steganography, the transfer domain provides better concealment of the secret image in the cover image, and has therefore proved much more reliable than spatial domain. In this paper, we attempt to maximize the retrieved secret PSNR against the original secret, while simultaneously minimizing the cover image degradation. This paper is built upon Discrete-Wavelet Transform to process the image while the Least Significant Bit method to store the information. We follow a principle of priority ordering the wavelet subspaces of both the secret and the cover with a view to make for the most efficient concealment. We propose the product of the secret and cover image PSNR and SSIM measures as the quantities to be maximized as it provides a more comprehensive evaluation of system performance, and study the performance against the choice of the number of levels of wavelet decompositions.

KEYWORDS

Cover; Secret; Stego; Embed; HAAR-DWT; LSB

1. INTRODUCTION

Labeling a message to be of high security will make it a high priority target for attacks. Likewise, enciphered messages always hold the risk of being discovered on route. Moreover, such messages can altogether be destroyed/tampered by a third party, if not decrypted successfully. Thus, the secrecy of transmission of such messages becomes important and here is where steganography takes over cryptography. Steganography, or image hiding, avoids overt declaration of the criticality of a message, by concealing the secret (image) in a mundane cover (image) so that its significance is known only to the intended recipient.

When the cover for embedding secret information is an image, the technique is referred to as Image Steganography. Any kind of signal can be stored into the cover; here, we hide a secret image: The secret image is embedded into the cover image resulting in the so called stego image. The recipient extracts the secret information out of stego image and gets the message. We use the Least Significant Bit (LSB) method to store the information, wherein the secret information is encoded in the least significant bits of the pixels of the cover image. The number of least significant bits used for this purpose varies as per application and the level of fidelity desired. Needless to say, a requirement of higher fidelity conflicts with a requirement of higher capacity.

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 11–24, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80502

In this paper, we compare two approaches: the first one using only two least significant bits of the cover to store the secret information while the other uses three. While the second provides more capacity, (i.e., more space), the first one leads to a higher cover-stego PSNR as well as SSIM.

In frequency domain approaches, both cover and secret are transferred to the transform domain and the transform coefficients of the secret are embedded in the transform coefficients of the cover to get a transform domain stego. This transform domain stego is converted back to the spatial domain by inverse transforms to get the spatial domain stego image. At the recipient end, to extract the secret information, one has to again perform the respective transform and extract the secret information from LSB of the coefficients. Also, the information of in how many and which coefficients the data is being embedded also needs to be recorded and transmitted. The conversion to the frequency domain helps because the human eye cannot detect changes in the high frequency while changes made in the low frequency can easily be detected. The concealment in the frequency domain spatially spreads the secret data over all the transform coefficients, preventing the retention.

The method used in this paper is a combination of LSB and iterative HAAR-DWT. We prioritize the space (the part where any changes can be least detected) in the cover image and energy in the secret image. Next comes the tradeoff between the amount of secret information being stored and similarity of the stego to the cover image. We observe results in terms of PSNR and SSIM, aiming for the most efficient solution, and state some problems that arise because of using HAAR-DWT.

2. BACKGROUND

The LSB method has been implemented after processing both the cover and secret images in various ways over the decades, for e.g., directly storing the secret image in LSB of cover image in spatial domain itself [1], performing DCT [2] or DWT [3] or DFT [4] on the cover image to convert it into frequency domain and then storing the information is LSB of respective coefficients.

Chandramouli and Memon [5] (2001) devised a method to calculate probability of detection in terms of number of bits hidden for storing information. Morkal, Tayana, et al [6] (2005) stated the applications and suitable uses of different steganographic techniques. Cheddad, Abbas, et al [7] (2010) have mentioned a state-of-the-art review and analysis of then existing steganography techniques. Similar work is done by Singh, Amritpal, et al. [8] (2014). Al-Korbi, Hamad A., et al [9] (2015) developed steganography technique storing information in the wavelet domain (RGB color). Vikas Patidar [3] (2016) developed a technique (monochrome/color) to store information in the HAAR-DDWT domain. We work on this base, performing HAAR-DWT iteratively on the image to better classify its wavelets as per the amount of information stored in them resulting in improved SSIM and PSNR performance. We also show there is a certain degree of permanent loss of information because of using HAAR-DWT with LSB method, making the entire process lossy. We state both the problem and reason of occurrence of this phenomenon.

3. HAAR-DWT

The HAAR wavelet is preferred because of its simple yet efficient decomposition process. It requires only simple addition/subtraction in horizontal and vertical directions to convert images from spatial to frequency domain. For a 4x4 matrix, the HAAR transform can be evaluated as follows:

$$\begin{bmatrix} a1 & b1 & a2 & b2 \\ c1 & d1 & c2 & d2 \\ a3 & b3 & a4 & b4 \\ c3 & d3 & c4 & d4 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} a1+b1+c1+d1 & a2+b2+c2+d2 & a1-b1+c1-d1 & a2-b2+c1-d2 \\ a3+b3+c3+d3 & a4+b4+c4+d4 & a3-b3+c3-d3 & a4-b4+c4-d4 \\ a1+b1-c1-d1 & a2+b2-c2-d2 & a1-b1-c1+d1 & a2-b2-c2+d2 \\ a3+b3-c3-d3 & a4+b4-c4-d4 & a3-b3-c3+d3 & a4-b4-c4+d4 \end{bmatrix}$$

Figure 1. Calculation of HAAR-DWT of an image.

Thus, the HAAR transform, applied once, decomposes any image into four frequency regions as $\begin{bmatrix} LL & HL \\ LH & HH \end{bmatrix}$, named low-low, high-low, low-high and high-high, allowing us to make changes in the high frequency region leaving the low frequency region containing the most significant information, untouched. Similar inverse operations can be made on the HAAR-transform to get back our original image.

4. EMBEDDING ALGORITHM

‘ns’ = number of times HAAR-DWT is desired to be done in secret, ‘x’ =(size of cover)/ (size of secret), where both these are square images and size means any one side, ‘nc’ =4^(n − (log2x) − 1), ‘N’ = number of final level secret image wavelets desired to be stored.

4.1 Hiding All Bits of Secret Information:

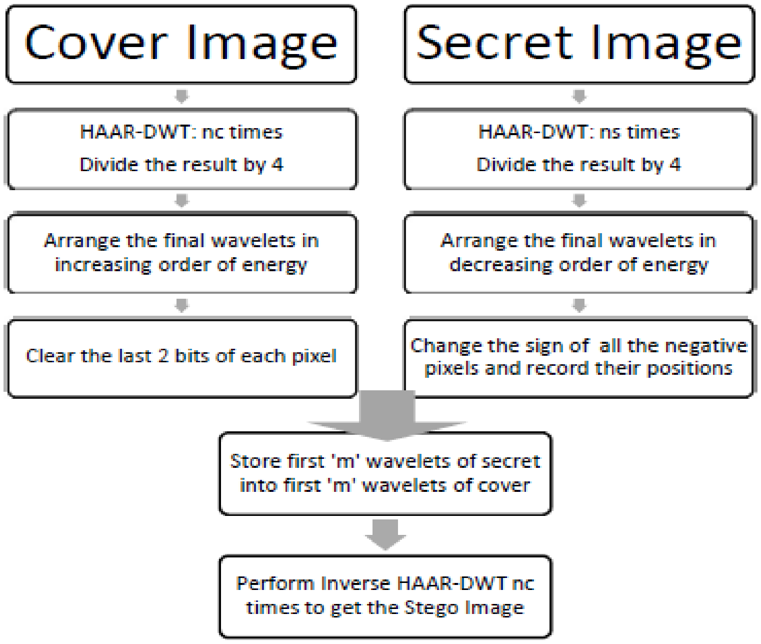

Cover Image
Secret Image
HAAR-DWT: nc times
Divide the result by 4
HAAR-DWT: ns times
Divide the result by 4
Arrange the final wavelets in increasing order of energy
Arrange the final wavelets in decreasing order of energy
Clear the last 2 bits of each pixel
Change the sign of all the negative pixels and record their positions
Store first 'm' wavelets of secret into first 'm' wavelets of cover
Perform Inverse HAAR-DWT nc times to get the Stego Image

Figure 2. (First) Embedding Algorithm

4.2 Hiding First Six Non-Zero Bits of Secret Information:

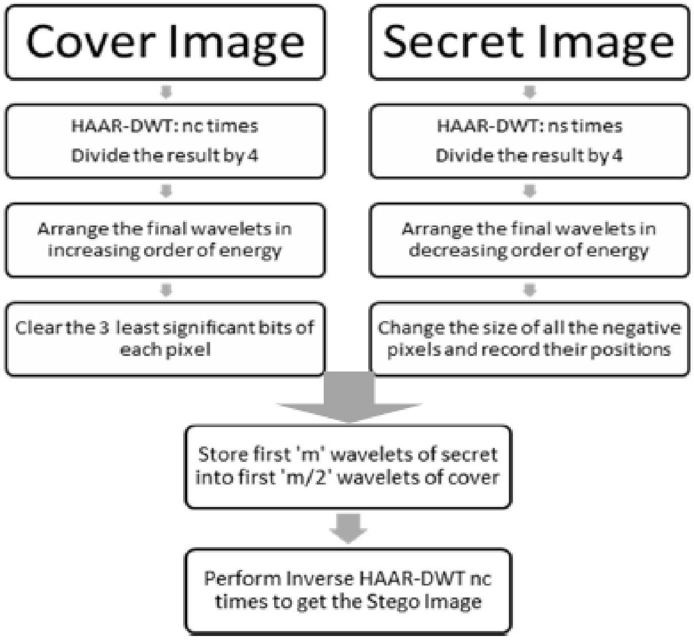

Cover Image
HAAR-DWT: nc times
Divide the result by 4
Arrange the final wavelets in increasing order of energy
Clear the 3 least significant bits of each pixel
Secret Image
HAAR-DWT: ns times
Divide the result by 4
Arrange the final wavelets in decreasing order of energy
Change the size of all the negative pixels and record their positions
Store first 'm' wavelets of secret into first 'm/2' wavelets of cover
Perform Inverse HAAR-DWT nc times to get the Stego Image

Figure 3. (Second) Embedding Algorithm

5. EXTRACTING ALGORITHM

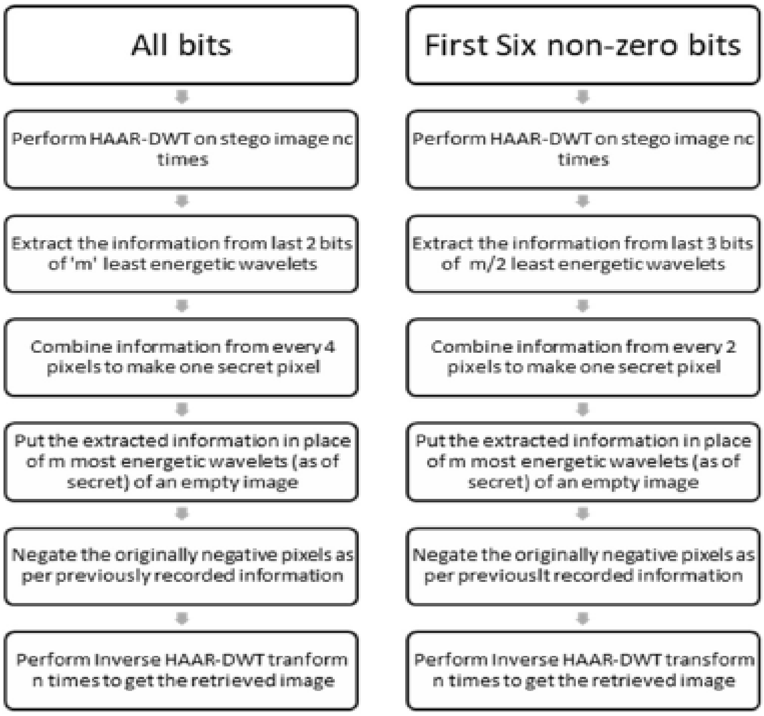

All bits
Perform HAAR-DWT on stego image nc times
Extract the information from last 2 bits of 'm' least energetic wavelets
Combine information from every 4 pixels to make one secret pixel
Put the extracted information in place of m most energetic wavelets (as of secret) of an empty image
Negate the originally negative pixels as per previously recorded information
Perform Inverse HAAR-DWT tranform n times to get the retrieved image
First Six non-zero bits
Perform HAAR-DWT on stego image nc times
Extract the information from last 3 bits of m/2 least energetic wavelets
Combine information from every 2 pixels to make one secret pixel
Put the extracted information in place of m most energetic wavelets (as of secret) of an empty image
Negate the originally negative pixels as per previouslt recorded information
Perform Inverse HAAR-DWT transform n times to get the retrieved image

Figure 4. Extracting Algorithm

6. OBSERVATIONS AND RESULTS

To decide the optimum value of ‘N’ for a given ‘n’, we study the product of the PSNRs and SSIMs of coverstego pair and secret-retrieved image pair [10][11]. The results are shown for n = 1, 2 and 3 (i.e. up to 3 levels of HAAR-DWT) each for both the cases of storing all bits and storing first six non-zero bits.

6.1 First Example

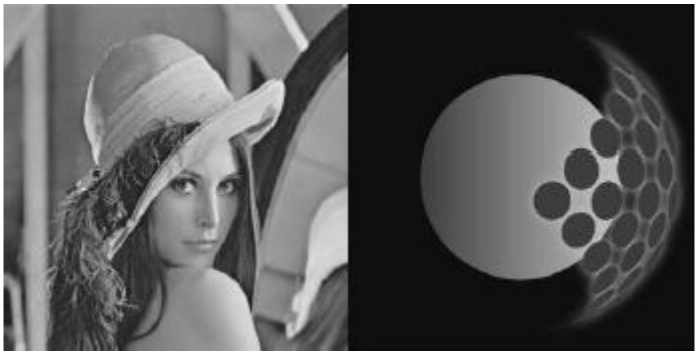


Figure 5. Cover (left) and Secret Image

6.1.1. ′n’ = 1

The PSNR and SSIM trend on storing N = 1: 4 wavelets of secret image (on horizontal axis):

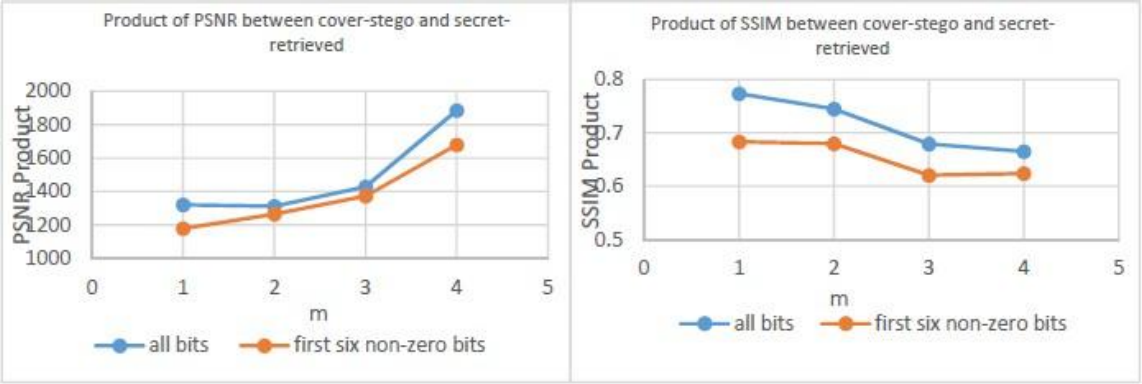


Figure 6. PSNR (left) and SSIM Product for ‘n’ = 1

6.1.2. ′n’ = 2

The PSNR and SSIM trend on storing N = 1: 16 wavelets of secret image (on horizontal axis):

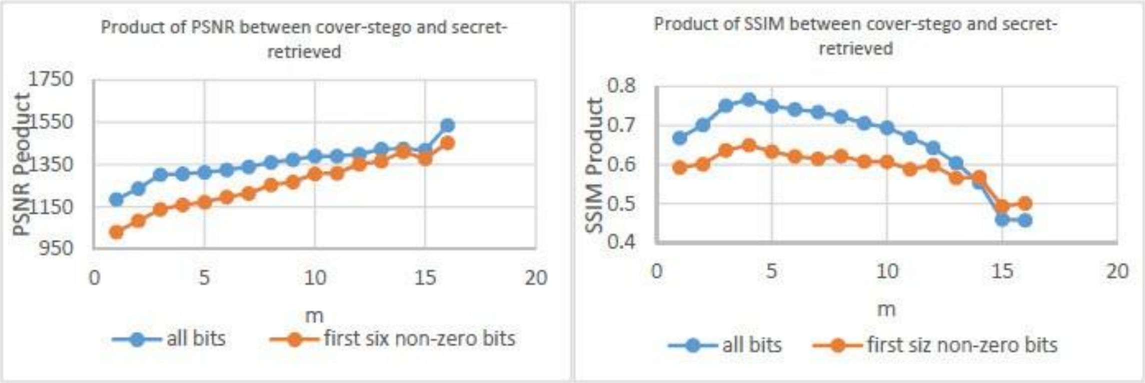


Figure 7. PSNR (left) and SSIM Product for ‘n’ = 2

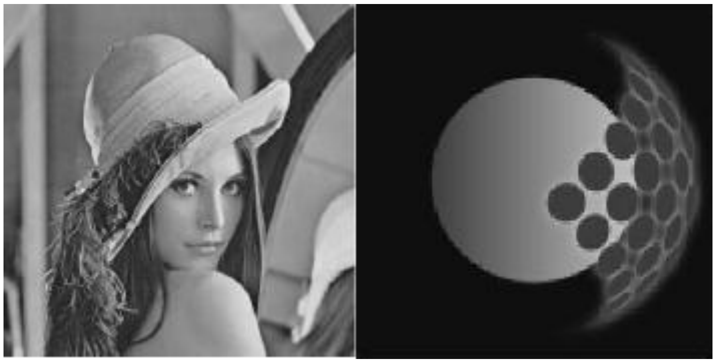


Figure 8. Stego (left) and Retrieved Secret Image (ɴ = 4)

6.1.3. ′n’ = 3

The PSNR and SSIM trend after storing m=1:64 wavelets of secret image (on horizontal axis):

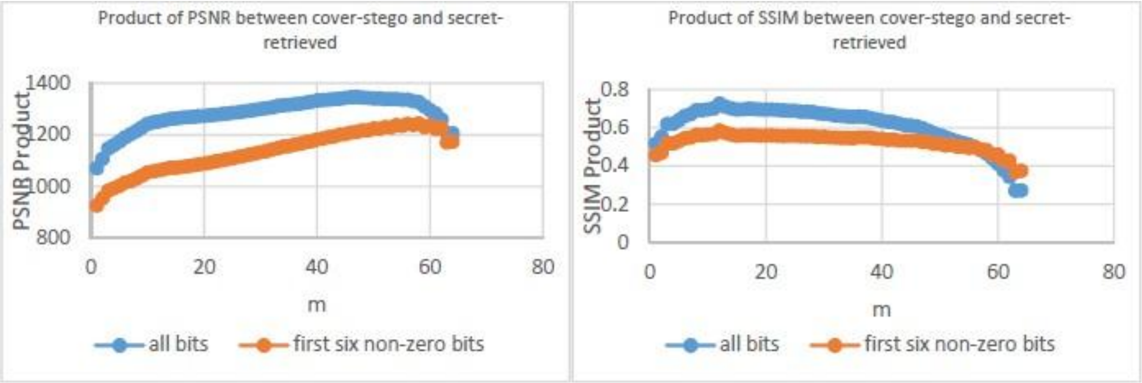


Figure 9. PSNR (left) and SSIM Product for ‘n’ = 3

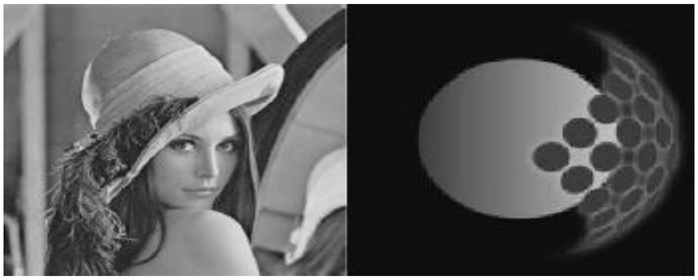


Figure 10. Stego (left) and Retrieved Secret Image (ɴ = 12)

Since the question here is of how similar the retrieved secret image is to the original secret image and how similar the stego image is to the cover image, we will follow the product of SSIM to get our optimum solution. Also, both PSNR and SSIM product are higher when we resort to storing all bits, as compared to storing only the first six non-zero bits in spite of increased storage capacity in the cover image. The results for optimum values of ‘ɴ’ (as per SSIM) are also shown in each sub-section above.

6.2 Second Example

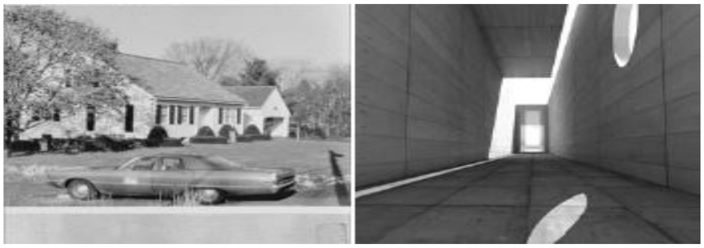


Figure 11. Cover (left) and Secret Image

6.2.1. ′n’ = 1

The PSNR and SSIM trend on storing ɴ = 1: 4 wavelets of secret image (on horizontal axis):

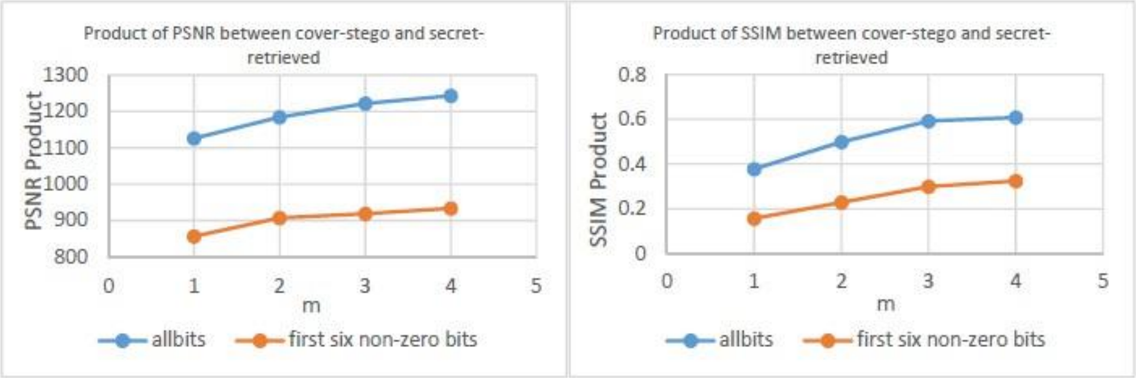


Figure 12. PSNR (left) and SSIM Product for ‘n’ = 1

6.2.2. ′n’ = 2

The PSNR and SSIM trend on storing ɴ = 1: 16 wavelets of secret image (on horizontal axis):

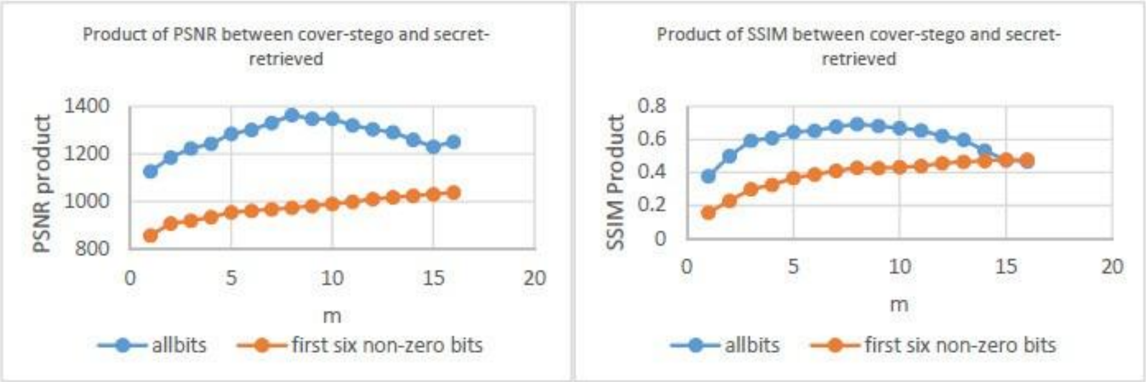


Figure 13. PSNR (left) and SSIM Product for ‘n’ = 2

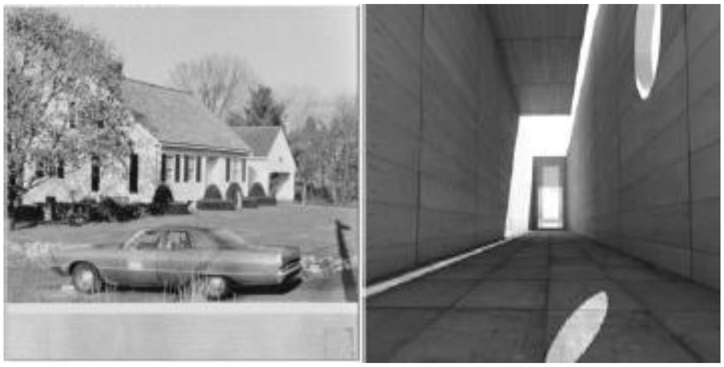


Figure 14. Stego (left) and Retrieved Secret Image (ɴ = 8)

6.2.3. ′n’ = 3

The PSNR and SSIM trend after storing m=1:64 wavelets of secret image (on horizontal axis):

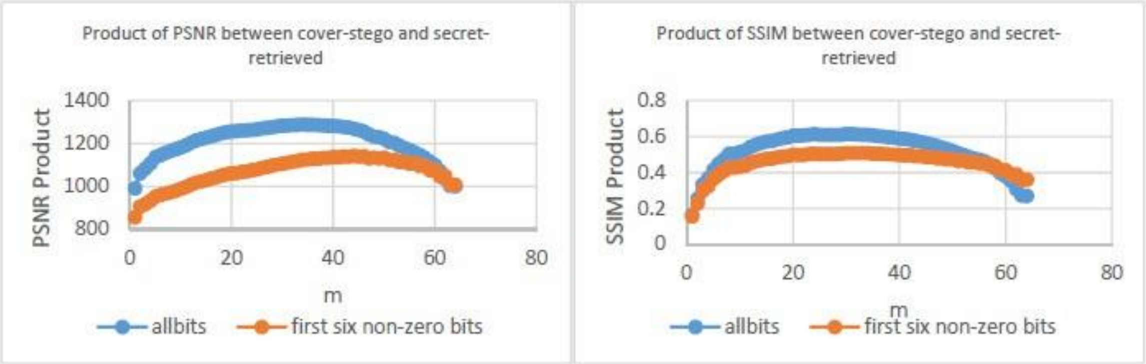


Figure 15. PSNR (left) and SSIM Product for ‘n’ = 3

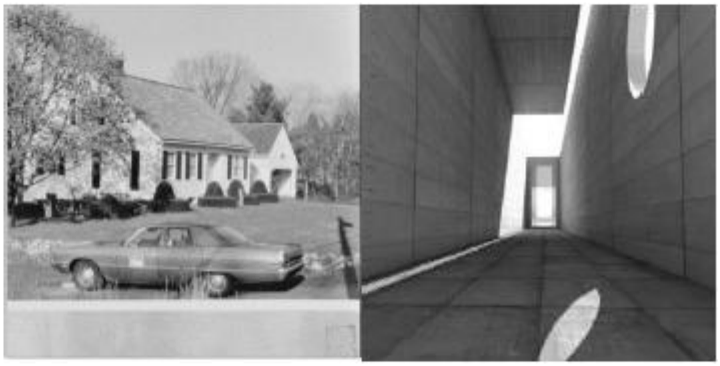


Figure 16. Stego (left) and Retrieved Secret Image (N = 28)

6.3 Third Example

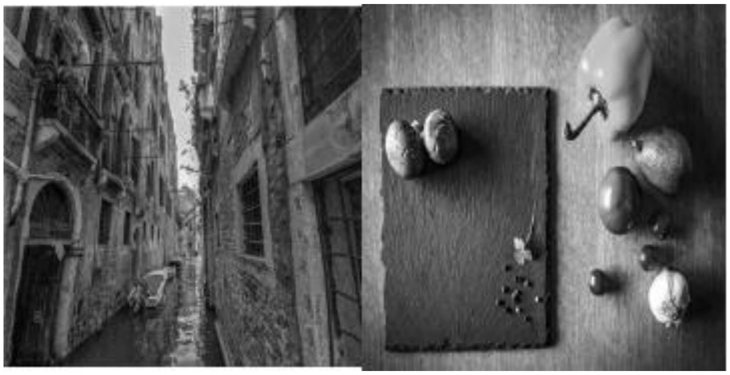


Figure 17. Cover (left) and Secret Image

6.3.1 ′n’ = 1

The PSNR and SSIM trend on storing N = 1: 4 wavelets of secret image (on horizontal axis):

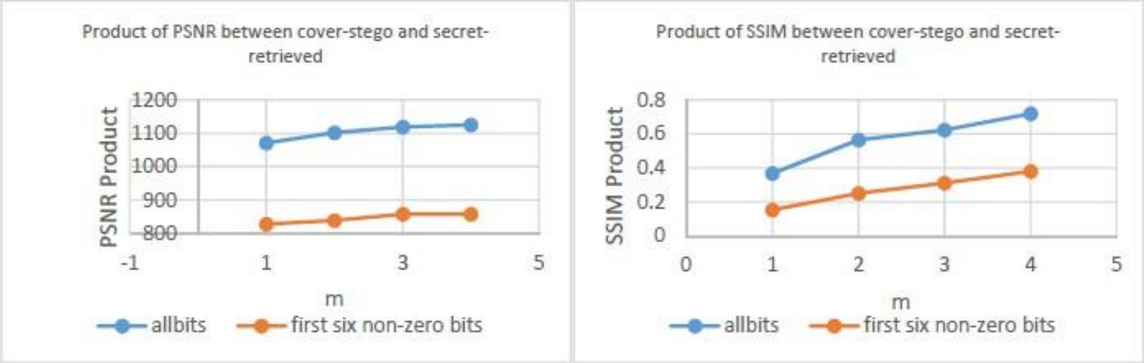


Figure 18. PSNR (left) and SSIM Product for ‘n’ = 1

6.3.2 ′n’ = 2

The PSNR and SSIM trend on storing ɴ = 1: 16 wavelets of secret image (on horizontal axis):

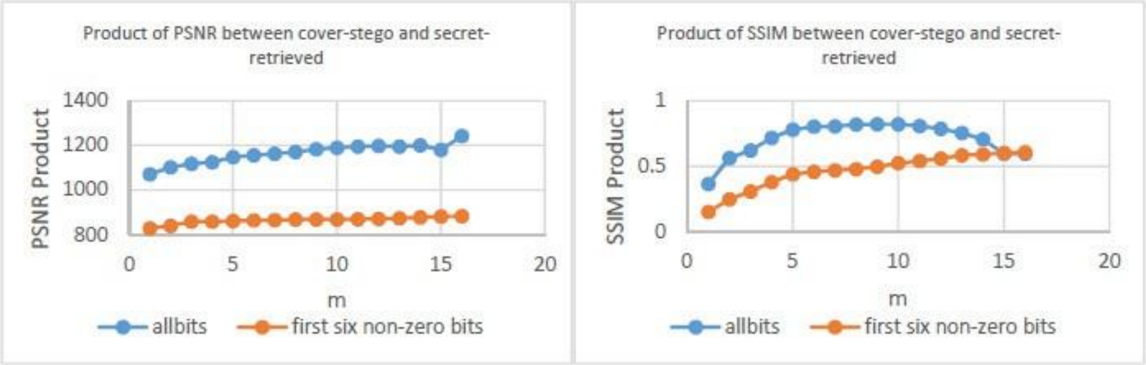


Figure 19. PSNR (left) and SSIM Product for ‘n’ = 2

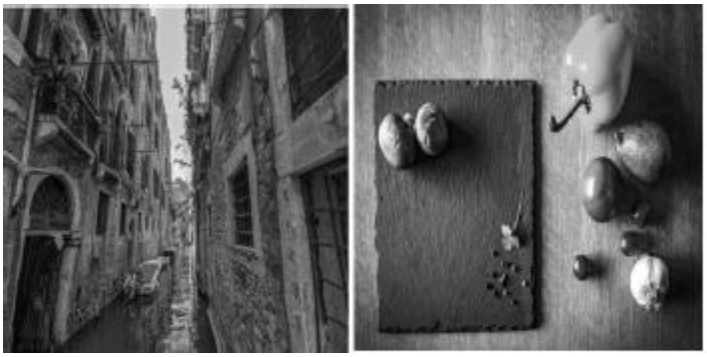


Figure 20. Stego (left) and Retrieved Secret Image (ɴ = 10)

6.3.3 ′n’ = 3

The PSNR and SSIM trend after storing m=1:64 wavelets of secret image (on horizontal axis):

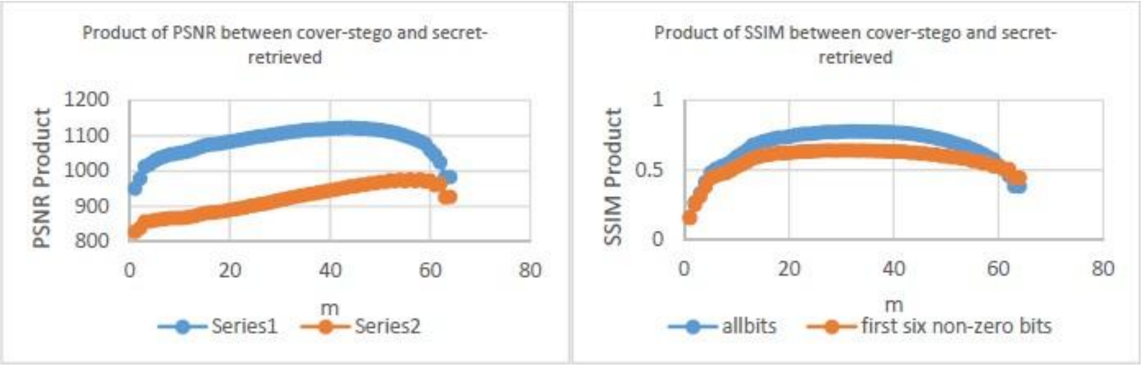


Figure 21. PSNR (left) and SSIM Product for ‘n’ = 3

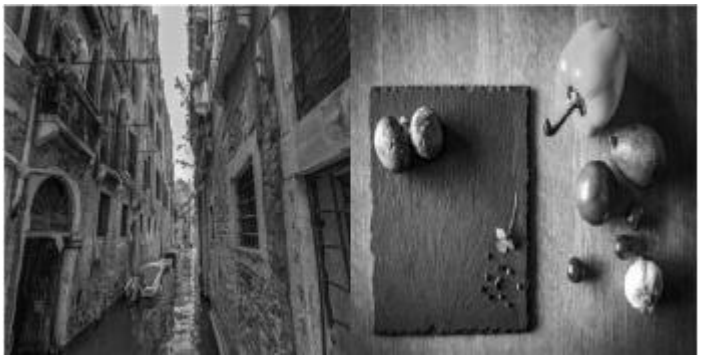


Figure 22. Stego (left) and Retrieved Secret Image (N = 35)

6.4 Average Result

We run the algorithm on a total of 10 cover-secret image pairs and plot the average results as follows:

6.4 1. 'n' = 1

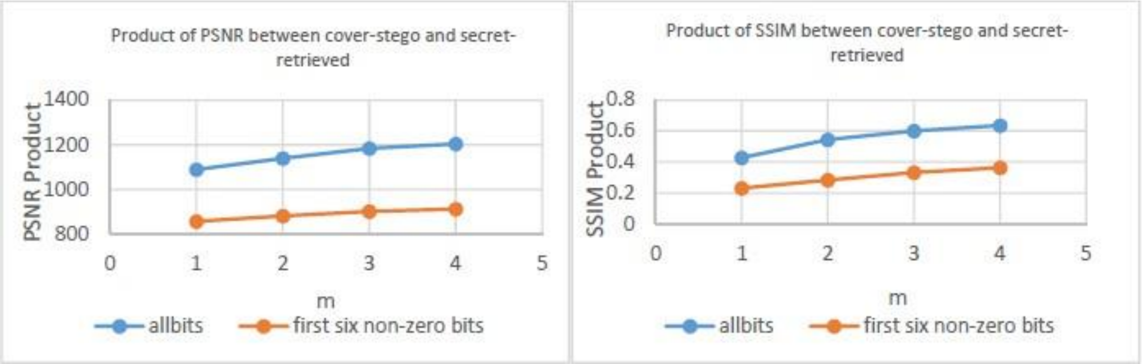


Figure 23. PSNR (left) and SSIM Product for 'n' = 1

6.4.2. 'n' = 2

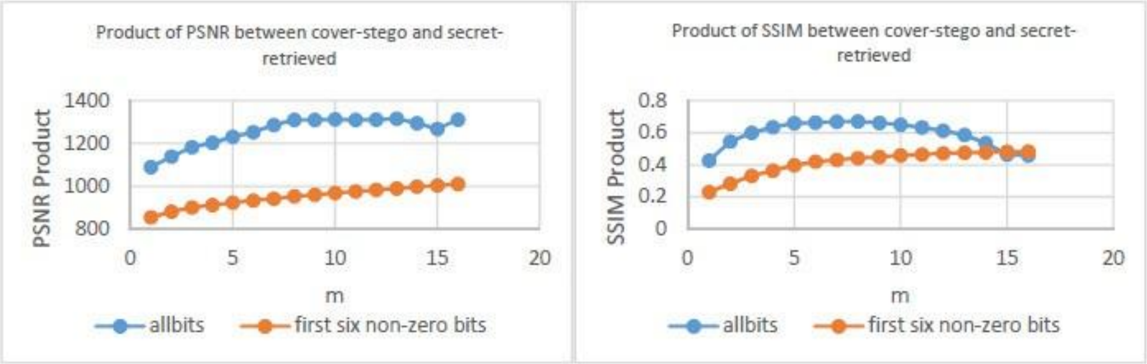


Figure 24. PSNR (left) and SSIM Product for 'n' = 2

6.4.3. ′n’ = 3

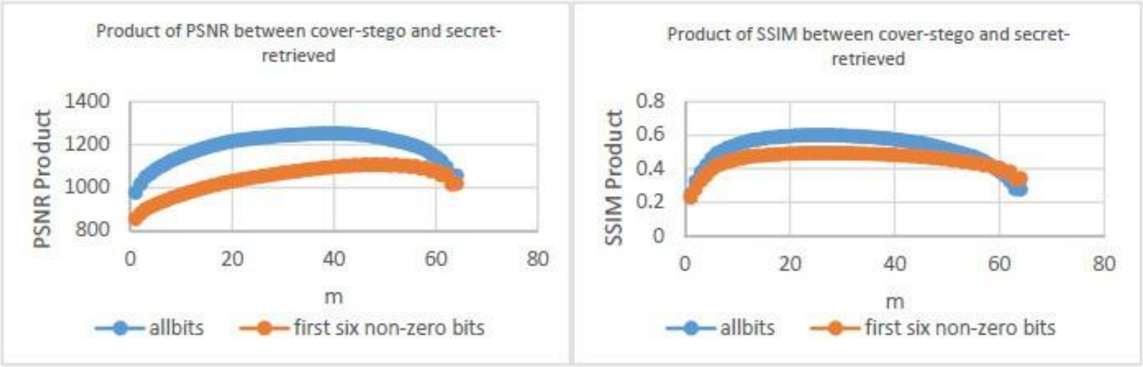


Figure 25. PSNR (left) and SSIM Product for ‘n’ = 3

7. SHORTCOMINGS

The procedure of evaluating the HAAR transform of an image includes addition as well as subtraction of pixel values as mentioned previously. This subtraction leads to some negative pixel values in the resulting wavelet transform which causes problems in both display of the image, and decimal to binary conversion required for application of the LSB method of embedding. To tackle this, one must record the position of all negative pixels in the transform and then temporarily assign them positive signs before converting them into binary for storing. After extracting the wavelets of the secret image from the stego, the sign of these pixel values should be restored before performing Inverse-HAAR to get the retrieved image.

The LSB method requires the pixel values to be in the range [0,255] for it to be convertible into 8-bit integers for further procedure. Consequently, we need to keep dividing the result by 4 every time we perform HAARDWT. This makes some of the resulting pixel values to be non-integers, which are subsequently rounded off when converted to 8-bits. This rounded-off information is lost forever. Therefore, even on storing the all the wavelets of the secret image, the retrieved secret doesn’t show a perfect SSIM of 1 with respect to the original secret. It can also be observed in many cases, but not all, that the PSNR product has started decreasing at the end of the third level which tells us not to go any further in HAAR-DWT levels (since, every time the size of each wavelet is becoming a fourth smaller with increasingly coarser approximation due to quantization error).

8. CONCLUSION

This paper presents a steganography technique using the LSB method. Encoding the secret image in transfer domain, rather than in spatial domain, and that too at different levels (i.e., after iteratively performing HAARDWT) with number of wavelets stored prioritizing their energy, we have found the optimum values of SSIM between cover-stego and secret-retrieved image by seeing the trend on varying the number of wavelets of the secret image being stored. We have also listed the shortcomings of this approach which increases the space complexity of the code, and besides increases the quantization error further. Like every other steganography technique, ours also has its advantages as well as shortcomings and can be fine-tuned as per the application it is to be used in.

REFERENCES

[1] R. K. Thakur and C. Saravanan, "Analysis of steganography with various bits of LSB for color images," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016

[2] K. Raja, C. Chowdary, K. Venugopal, and L. Patnaik, "A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images," 2005 3rd International Conference on Intelligent Sensing and Information Processing, 2005.

[3] Vikas Patidar, "Techniques of Image Concealment," M.Tech Thesis, Indian Institute of Technology Kanpur, Kalyanpur, Uttar Pradesh, India, 2016.

[4] D. Bhattacharyya and T.-H. Kim, "Image Data Hiding Technique Using Discrete Fourier Transformation," Communications in Computer and Information Science Ubiquitous Computing and Multimedia Applications, pp. 315–323, 2011.

[5] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205).

[6] Morkel, T., Eloff, J. H., & Olivier, M. S. 2005, "An overview of image steganography," ISSA, pp. 1-11, 2011.

[7] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: Survey and analysis of current methods," Signal Processing, vol. 90, no. 3, pp. 727–752, 2010.

[8] Singh, A., & Singh, S. J. 2014, "An Overview of Image Steganography Techniques," International Journal of Engineering and Computer Science, vol3, (7), 7341-7345, 2014.

[9] H. A. Al-Korbi, A. Al-Ataby, M. A. Al-Taee, and W. Al-Nuaimy, "High-capacity image steganography based on Haar DWT for hiding miscellaneous data," 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2015.

[10] SIPI Image Database. [Online]. Available: http://sipi.usc.edu/database/. [Accessed: 07-Feb-2018].

[11] "Free high quality photos · Pexels," Free Stock Photos. [Online]. Available: https://www.pexels.com/. [Accessed: 07-Feb-2018].

[12] "Bilinear interpolation Definition from PC Magazine....", [Online]. Available: https://www.pcmag.com/encyclopedia/term/38607/bilinear-interpolation. [Accessed: 5- Feb- 2018].

[13] "Understanding Digital Image Interpolation", Cambridgeincolour.com, 2018. [Online]. Available: https://www.cambridgeincolour.com/tutorials/image-interpolation.htm. [Accessed: 05- Feb-2018].

[14] A. Bogomolny, "Equations of a Straight Line from Interactive Mathematics Miscellany and Puzzles", Cutthe-knot.org, 2018. [Online]. Available: https://www.cut-theknot.org/Curriculum/Calculus/StraightLine.shtml. [Accessed: 05- Feb- 2018].

[15] "High-Resolution Antialiasing|NVIDIA", Nvidia.com, 2018. [Online]. Available: http://www.nvidia.com/object/feature_hraa.html. [Accessed: 05- Feb- 2018].

[16] “Hardware Knowledgebase - What is supersampling (antialiasing technique)? - HardwareFAQs: powered by neofaq”, Web.archive.org, 2018. [Online]. Available: https://web.archive.org/web/20060325144730/http://www.neoseeker.com/Hardware/faqs/kb/10,72.html. [Accessed: 05- Feb- 2018].

[17] “Supersampling - Everything2.com”, Everything2.com, 2018. [Online]. Available: http://www.everything2.com/index.pl?node_id=1028947. [Accessed: 05- Feb- 2018].

[18] P. Getreuer, “Image Interpolation with Contour Stencils”, 2018. [Online]. Available: http://www.ipol.im/pub/art/2011/g_iics/. [Accessed: 05- Feb- 2018].

AUTHORS

Aditi Singh – Undergraduate Student, IIT Kanpur; Research Interests - Image and Video Processing, Computer Graphics. Webpage – http://home.iitk.ac.in/~aditisgh/

K S Venkatesh – Professor, IIT Kanpur; Research Interests - Signal, Image and Video Processing with applications in Computer Vision, Machine Vision, Computational Photography and Medical Imaging; Robot Navigation. Webpage - http://home.iitk.ac.in/~venkats/

Vikas Patidar – Former Master’s Student, IIT Kanpur

FPGA-IMPLEMENTATION OF WAVELET-BASED DENOISING TECHNIQUE TO REMOVE OCULAR ARTIFACT FROM SINGLE- CHANNEL EEG SIGNAL

Chen Ronghua, Li Dongmei and Zhang Milin

Department of Electronic Engineering, Tsinghua University, Beijing, China

ABSTRACT

This paper presents the real-time implementation on FPGA of the wavelet-based denoising technique to remove the ocular artifact from the signal-channel EEG signal. The advantage of this method over conventional methods is that there is no need for the recording of the electrooculogram (EOG) signal itself. This approach papers both for eye blinks and eye movements. Discrete Wavelet Transform (DWT) is selected end the hard-thresholding is applied to the wavelet coefficients using the Statistical Threshold (ST) estimated in interested bands. This real-time architecture presents two characteristics: 1) quantization of the filter coefficients and the elimination of the multiplier to reduce the hard cost, and 2) symmetrical extension of the signal boundary to full reconstruction while the data volume is invariable. Experimental results show that proposed architecture efficiently removes the ocular artifact from EEG signal.

KEYWORDS

Wavelet transform, EEG, ocular artefact, hard-thresholding, denoising

1. INTRODUCTION

Electroencephalogram (EEG) is the recording of the brain's neuronal activity by placing electrodes on the scalp [1]. The EEG remains most of cerebral information that has been utilized in many medical diagnosis and therapies including epilepsy, sleep disorder, and so on [2]. However, EEG records are often corrupted by different types of artifacts that lead to the requirement of the use of complex methods for identification and to an increase in the difficulty in analysing the clinical information. Therefore, artifact removal is very necessary. Those artifact sources are usually divided into two categories: extra physiologic, such as power-line interference and electrodes noise, and physiologic, such as eye, muscle, and cardiac activities. The former artifact can often be removed by traditional filtering techniques, but removal of the latter artifact requires careful attention due to the fact that it can be within the same frequency range of the EEG signal [3]. Compared with other physiologic artifacts, the ocular artifact is the most significant, so a real-time ocular artifact removal technique is our aim.

Ocular artifact is caused by eye movements and eye blinks during the EEG recording and have frequency ranges of 0–7 Hz and 8-13 Hz, respectively. But sometimes, vertical eye movement

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 25–35, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80503

artifacts seem to produce a rise in the higher frequencies [4]. Therefore, we applied threshold denoising in bands with frequency between 0-16Hz. The widely used methods for ocular artifact removal from EEG signal are based on the time domain and frequency domain like Principal Component Analysis (PCA) [5] and the Independent Component Analysis (ICA) [6], which are also shown to be efficiently to remove ocular artifact, but they rely on multiple channel data. Fourier transform (FT) [7] and short-time Fourier transform (STFT) [8] have already been applied for signal analysing but they are also suffering from shortcomings when handling non-stationary and non-deterministic EEG signal. From the variety of approaches available, the Wavelet transform (WT) was found to be the most effective time-frequency domain analysis method to deal with EEG signal, because WT provides accurate frequency information at low frequencies and accurate time information at high frequencies, and this property is matched with biomedical applications [9].

In this paper, we implement the wavelet-based denoising algorithm to real-time removal the ocular artifact from the single-channel EEG signal and verify hardware designs on Xilinx FPGA.

2. WAVELET-BASED DENOISING

2.1. Discrete Wavelet Transform

The continuous wavelet transform (CWT) of a signal $x(\tau)$is defined as the correlation between $x(\tau)$and the basis function $\psi_{(\alpha,\beta)}(\tau)$ as follows:

$$\mathrm{CWT}_{(\alpha,\,\beta)}=\int_{-\infty}^{+\infty}\chi(\tau)\,\psi^{*}_{(\alpha,\,\beta)}(\tau)\;d\tau \tag{1}$$

Where (*) denotes the complex conjugate and $\psi_{(\alpha,\beta)}(\tau)$ are obtained by performing dilations and shifting of the mother wavelet $\psi(\tau)$.

$$\Psi_{(\alpha,\,\beta)}=1/\sqrt{\alpha}\psi(\tau-\beta/\alpha) \tag{2}$$

Where α, β is called the scale factor and the time translation factor. The scale factor to approximate different frequencies by compression or stretching, the time translation factor enables the wavelet traverse the signal. A large value of scale parameters represents analysis of low-frequency components of the signal. On the other hand, a small value of this parameters represents analysis of high-frequency components of the signal.

Because of the continuous values of α and β, he CWT has a lot of redundancy in computation, which is not what we want. When α, β is selects as discrete numbers that defined on the basis of power of two as follows:

$$\alpha_{\lambda}=2^{\lambda} \tag{3}$$

$$\beta_{\lambda,\,\kappa}=2^{\lambda}k,\;\;\lambda,\,k\in Z \tag{4}$$

Then DWT is obtained and defined as follows:

$$\mathrm{DWT}_{(\lambda,\,k)}=2^{-\lambda/2}\int_{-\infty}^{+\infty}\chi(\tau)\,\psi^{*}\left(2^{-\lambda}\tau-k\right)d\tau \tag{5}$$

When α is selects as (3), but β is still continuous values, then stationary wavelet transform (SWT) is obtained.

$$\mathrm{SWT}_{(\lambda,\,\kappa)}=2^{-\lambda/2}\int_{-\infty}^{+\infty}\chi(\tau)\,\psi^{*}\left(2^{-\lambda}\tau-2^{-\lambda}\kappa\right)d\tau \tag{6}$$

Compare (1), (2), (5) and (6), DWT with orthogonal wavelet is considered non-redundant and highly efficient wavelet transform to obtain discrete wavelet representation of signals, it requires less computational resources than others wavelet transform for real-time analysis. Therefore, in this paper, we choose DWT, which is a good choice for single-channel hardware implementation.

2.2. Discrete Wavelet Transform

Mallat algorithm [10] is usually used to compute the DWT, it can speed up the calculation of DWT. This algorithm is based on a pair of low-pass (H) and high-pass (G) filters, named quadrature mirror filters (QMF), their relationship as follows:

$$\mathrm{G(n)}=(-1)^{k}\mathrm{H(N-n-1)}. \tag{7}$$

Where N is the number of filter coefficients.

These filters are constructed from the wavelet function ψ(τ) and scaling function φ(τ), their relationship as follows:

$$\varphi(\tau)=\sqrt{2}\sum\nolimits_{K}\mathrm{H(k)}\varphi(2\tau-\mathrm{k}). \tag{8}$$

$$\psi(\tau)=\sqrt{2}\sum\nolimits_{K}\mathrm{G(k)}\psi(2\tau-\mathrm{k}). \tag{9}$$

The outputs of the high-pass filters corresponds to the high frequency components of the signal, called details components $d_{\lambda}(k)$ and the outputs of the low-pass filters corresponds to the low frequency components of signal, called approximations components $a_{\lambda}(k)$. Using the $d_{\lambda}(k)$ and $a_{\lambda}(k)$ can fully reconstruct original signal, this process is called the inverse discrete wavelet transform (IDWT).

The IDWT used a pair of low-pass $(\tilde{\mathrm{H}})$ filters and high-pass filters $\tilde{\mathrm{G}}$ to reconstruction. The decomposition and reconstruction filters are related to each other as follows:

$$\tilde{\mathrm{H}}=\mathrm{H(N-n-1)} \tag{10}$$

$$\tilde{\mathrm{G}}=\mathrm{G(N-n-1)} \tag{11}$$

There, four filters have the same absolute value, but sign and position are different from each other.

2.3. Threshold and Thresholding Function

The wavelet-based denoising technique are usually based on the Donoho algorithm [11] that consists to apply a thresholding function to the wavelet coefficients at different scales. The main idea of this algorithm is to compared the wavelet coefficients with the preset threshold.

Hard thresholding function as follows:

$$\omega = \begin{cases} \omega & |\omega| \geq T \\ 0 & |\omega| < T \end{cases} \tag{12}$$

Soft thresholding function as follows:

$$\omega = \begin{cases} \operatorname{sgn}(\omega)(|\omega| - T) & |\omega| \geq T \\ 0 & |\omega| < T \end{cases} \tag{13}$$

Where ω is the wavelet coefficients, T is the preset threshold.

A coiflet 3 wavelet (coif3) filter has been chosen, since the shape of its mother wavelet resembles the shape of the eye blink artifact [4]. According to [3], [12], the statistical threshold T with hard thresholding function would be better, T as follows:

$$T = 1.5\sigma(Hk)$$

Where σ (Hk) is standard deviation of detail coefficients at the k level.

2.4. Performance Metrics

Signal to artifact ratio (SAR), signal to noise ratio (SNR) and root mean square error (NMSE) are used in this paper to evaluate denoising performance [3], [13]. SAR, SNR and NMSE as follows:

$$\text{SAR}=10\log(\sigma(\chi)/\sigma(\chi - \tilde{\chi})). \tag{15}$$

$$\text{SNR}=10\log(\textstyle\sum_{\lambda=1}^{k} \chi(\lambda)^2 / \sum_{\lambda=1}^{\kappa} (\chi(\lambda) - \tilde{\chi}(\lambda))^2 \tag{16}$$

$$\text{RMSE}=\sqrt{\textstyle\sum_{\lambda=1}^{\kappa} (\chi(\lambda) - \tilde{\chi}(\lambda))^2 / k} \tag{17}$$

Where χ is original signal, $\tilde{\chi}$ is denoising signal, and k is the length of χ.

3. Implementation and Analyse

In order to implement the method of using DWT to remove the ocular artifact with Mallat algorithm and Donoho algorithm on hardware, we need quantization of the filter coefficients and overcome boundary effects. There, pipeline technique and symmetrical-extendsion technique are used to solving filters implement and avoid boundary effects.

3.1. EEG Data Source

The EEG data is used in this paper are taken from the BCI Competition 2008 Graz data set B, was publicly available [14]. This database consists of EEG data from 9 subjects, recorded the real EEG signal and electrooculogram (EOG) signal. All of data were recorded with a sampling frequency of 250 Hz, and they were through a band-pass filter between 0.5 Hz and 100 Hz, and a notch filter at 50 Hz to filter power-line noise.

The eye and brain activities have physiologically separate sources, it makes them independent [15], and can be represented as follows:

$$EEG_{rec} = EEG_{true} + k \cdot \mathrm{EOG} \tag{18}$$

Where k represents the propagation factor. According to [18], we can consider EEGtrue is clean EEG signal, and EEGtest is contaminated EEG signal by ocular artifact. We used EEGtest as we need for this paper. In this paper, k=1.

3.2. Implementation Architecture

The proposed architecture consists of three parts: decomposition, denoising and reconstruction. That is based on the Mallat algorithm and Donoho algorithm described in the previous mentioned. As shown in Figure 1, for acceptable computational complexity and obtain the frequency range of interest, the contaminated EEG decomposition five level, the correspond frequency down to up almost in 0-4 Hz, 4-8 Hz, 8-16 Hz, 16-32 Hz, 32-64 Hz and 64-128 Hz.

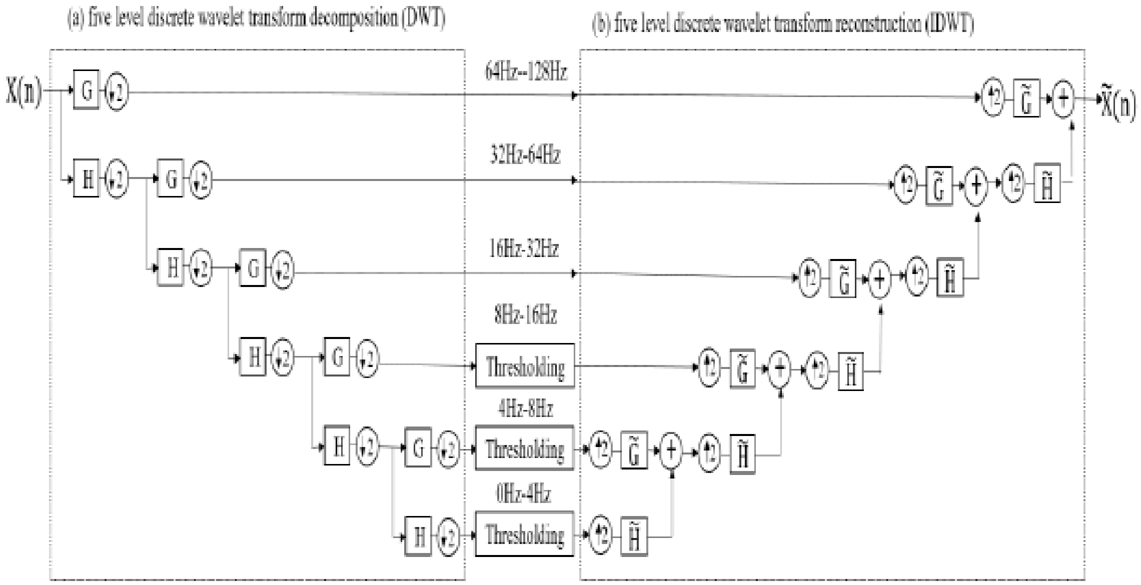

(a) five level discrete wavelet transform decomposition (DWT)
(b) five level discrete wavelet transform reconstruction (IDWT)
X(n)
G
H
↓2
64Hz--128Hz
32Hz-64Hz
16Hz-32Hz
8Hz-16Hz
4Hz-8Hz
0Hz-4Hz
Thresholding
Thresholding
Thresholding
↑2
G̃
H̃
+
X̃(n)

Figure 1. Real-time denoising architecture based on wavelet transform

When a sequence x(n) coming, x(n) is decompose by DWT into detail components and approximation components, and double sampling. Then only the approximations components are continue divided into two part as before. The last two level include one approximations component and two detail components are denoising by hard thresholding to get the new wavelet coefficients. When the decomposition and denoising are done, insert one zero between each sample. Then used the new wavelet coefficients to reconstruction the sequence, the process is just the opposite of the decomposition.

3.3. Filters Coefficients

In order to implement the DWT filters on hardware, the filters coefficients should be quantified that is enlarges the filter coefficients and then take the integer par. In this paper, 32 times, 64 times, 128 times and 256 times are applied to test EEG signal, the reconstruction results show that enlarge 256 times can bring minimum error that we can allowed, as shown in the Figure 2 (a), (b), (c), (d). Obviously the higher times error will be lower, but it will lead to greater hardware cost.

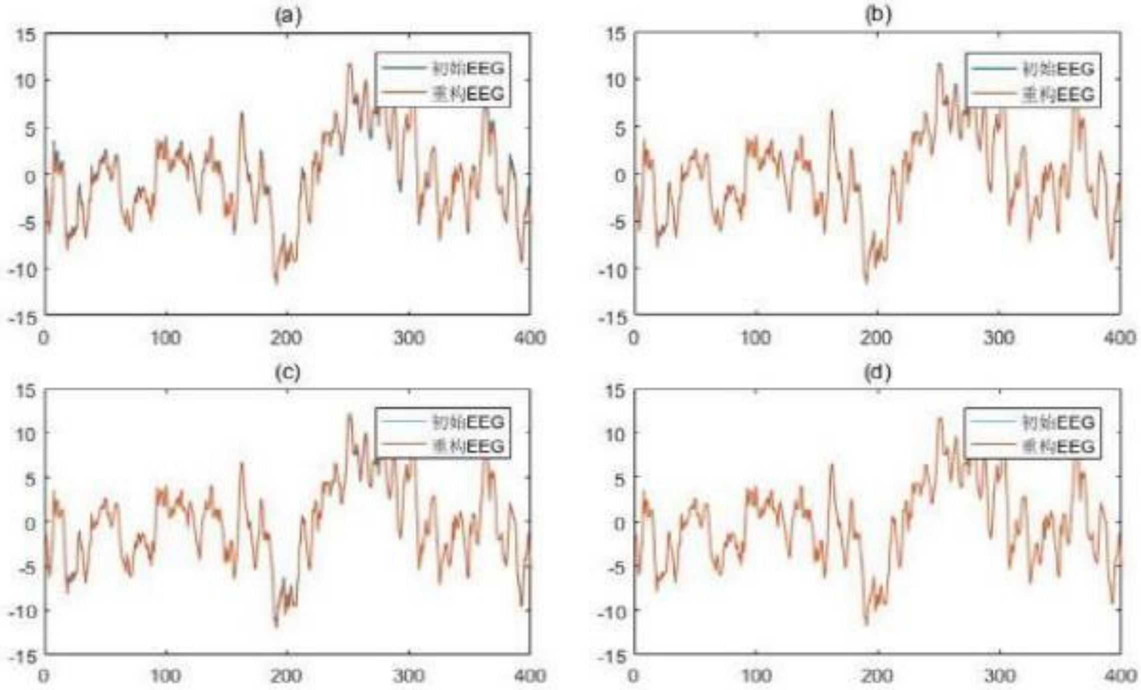


Figure 2. EEG reconstruction results with different quantification

Then used the power of two to represent the coefficients after expand 256 times, taking coefficients about colif3 filters (H) for example, as shown in the Table 1.

According to this characteristic, we applied shift-adder replaces the multiplier to implement no multiplier FIR filter, and take account of quantification, four coefficients are going to zero, so that only fourteen coefficients are useful data, four level pipeline can complete the addition. Because of the DWT filters features, just like the previous analysis four kinds of filters (H, G, $\tilde{H}$,$\tilde{G}$) can be used this structure ,as shown in Figure 3.

3.4. Boundary Effect

DWT is assuming that the data is infinite, but in practice, the data is often limited, so overcome boundary effect is necessary. According to the DWT algorithm, it is a convolution process. So when used filters to replace convolution to deal with limited data will bring boundary effects, especially the filter length is longer. There are usually three ways to extend the boundary: zero-extension, cycle-extension and symmetrical-extension.

Zero-extension is used zero instead of data which beyond the boundary, it's advantage is simple, but will lead to distortion duo to mutation. Cycle-extension is periodization of the original signal, it will increase a lot of computation. Symmetrical-extension only used boundary data, and can be fully reconstruction without adding data.

To avoid the boundary effect, in this paper choose extend seventeen boundary data that symmetric with the original boundary data before data through decomposition filters. Then select valid data to reconstruction enable the data volume is invariable, this equivalent to reducing the amount of reconstruction computation, it is important for hardware implementation. Due to the five-level is the same architecture to one-level, so there take one-level denoising based on DWT-IDWT with coif3 wavelet for example, as shown in Figure 4. Where L is the length of the data in the corresponding position, the original data length set 32, and the outputs data is select last 32 data from 49. Experiments show that it can be fully reconstructed. So in the five-level architecture, when data enter into second level, the data is not 49 but 32, following level is the same as this. Finally, a lot of computation is reduced.

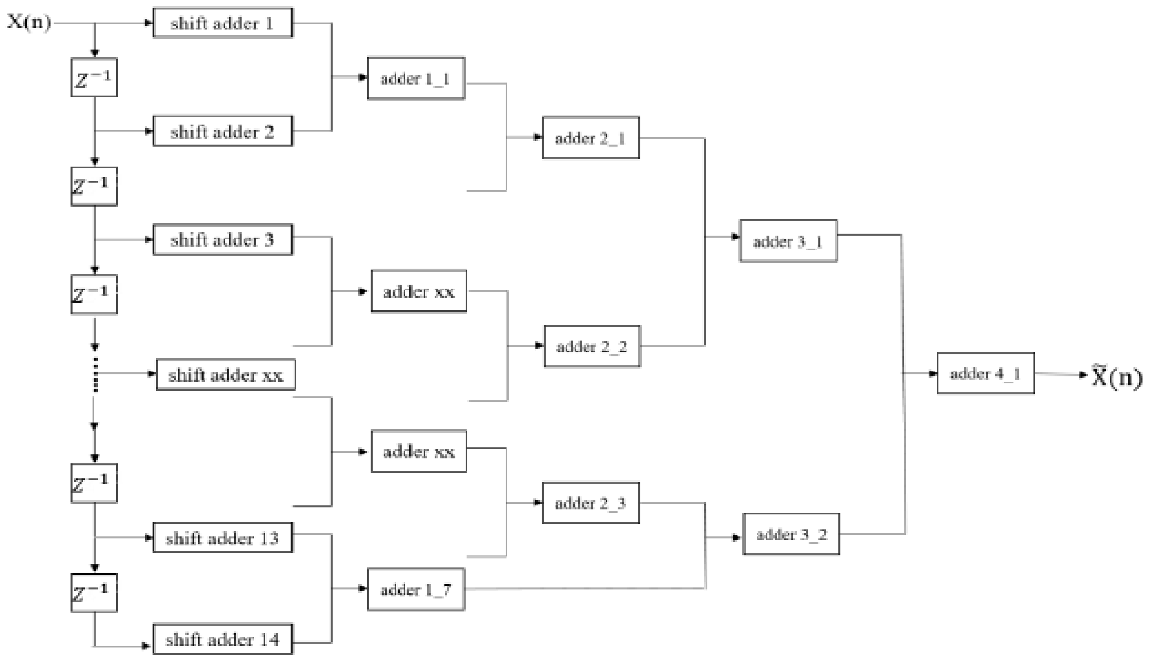


Figure 3. Shift adder and four level pipeline addition structure used to implement filters

Table 1 Filter coefficients (H) of coif3 wavelet

Coefficients(H)	Original	Trunc	Power of two
1	0.00379351286	1	2^0
2	0.00778259642	2	2^1
3	-0.02345269614	-6	-2^2-2^1
4	0.06577191128	-17	2^4-2^0
5	0.06112339000	16	2^4
6	0.40517690240	104	$2^7-2^4-2^3$
7	-0.79377722262	-203	$-2^8+2^6-2^3-2^1-2^0$
8	0.42848347637	110	$2^7-2^4-2^0$
9	0.07179982161	18	2^4+2^1
10	-0.08230192710	-21	$-2^4-2^2-2^0$
11	-0.03455502757	-9	-2^3-2^0
12	0.01588054486	4	2^2
13	0.00900797613	2	2^1
14	-0.00257451768	-1	-2^0
15	-0.00111751877	0	0
16	0.00046621696	0	0
17	0.00007098330	0	0
18	-0.00003459977	0	0

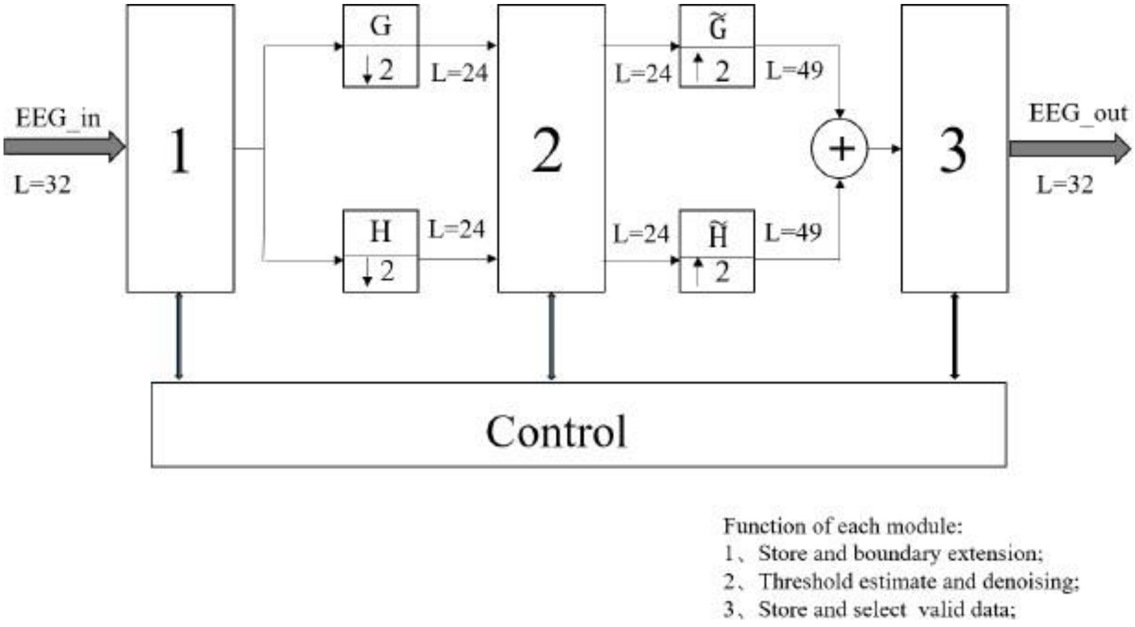

EEG_in
L=32
1
G
↓2
L=24
H
↓2
L=24
2
L=24
G̃
↑2
L=49
L=24
H̃
↑2
L=49
+
3
EEG_out
L=32
Control
Function of each module:
1、Store and boundary extension;
2、Threshold estimate and denoising;
3、Store and select valid data;

Figure 4. FPGA architecture for one-level denoising based on DWT-IDWT with coif3 wavelet

3.5. FPGA Results

We used previous EEG database to obtain EEG test data by formula (18) for this paper. In this paper, SAR, SNR and RMSE to be counted before and after denoising, shown in Table 2. The results shown that SNR have big improvement. The greater value of SAR, is considered to the more artifact is removed, and the smaller value of RMSE is considered to before and after artifact denoising the error is smaller, which means that the useful signal is retained as far as possible. In Figure 5, it also shows that the ocular artifact is effective removal from EEG signal. Besides, we can find the ocular artifact maximum frequency is over 13Hz, it fits with what we said before.

Table 2. Indication of EEG before and after denoising

Indication	Before	After	Difference Value
SNR	-51.53	1.28	53.21
SAR	-25.96	-9.98	15.98
RMSE	53.32	51.92	-1.4

4. CONCLUSIONS

In this paper, a real-time removal of the ocular artifact from signal-channel EEG signal based on discrete wavelet transform was implemented on FPGA. In addition to considering hardware cost in the implementation, boundary effects are also take into account. The method what we adopted to remove ocular artifact shows effective, and reduced a lot of computation.

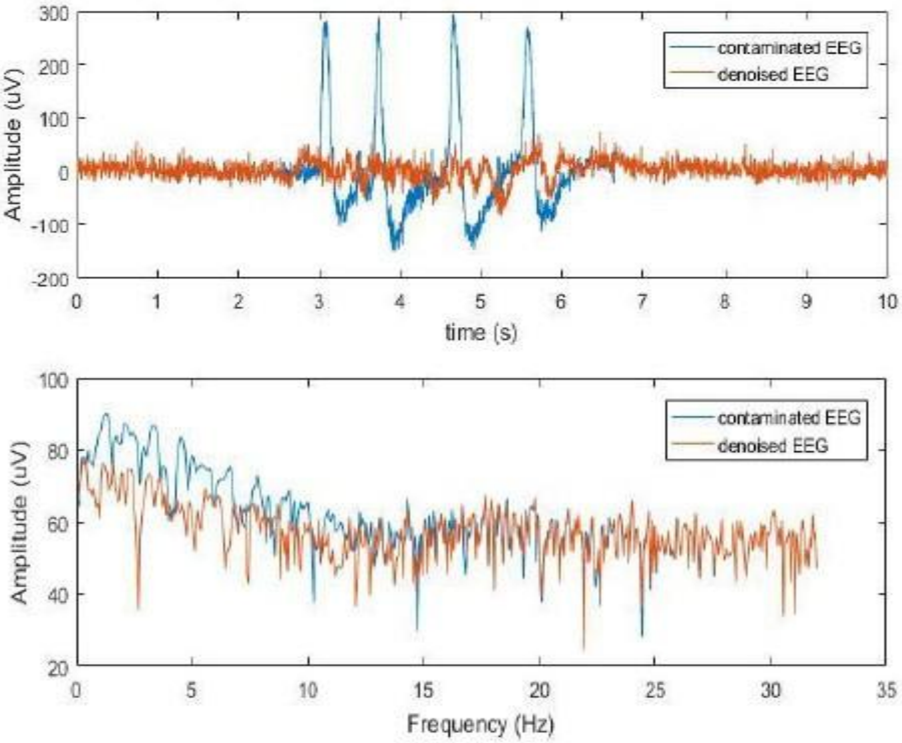


Figure 5. Compare of EEG before and after denoising

In the future, we will apply this denoising structure in the Brain Machine Interface (BCI) system to compare the effects about feature extraction and classifications for EEG signal before and after denoising.

ACKNOWLEDGEMENTS

This paper was supported by Prof. Li and Prof. Zhang, and my friends in the lab. Thanks you for all of people who helped me.

REFERENCES

[1] Tatum, W.O. & Husain, A.M. & Benbadis, S.R. &Kaplan, P. W.,(2007) "Handbook of EEG Interpretation", New York, NY, USA: Demos Medical Publishing.

[2] Sakkalis, V, (2011) "Review of advanced techniques for the estimation of brain connectivity measured with EEG/MEG", Comput. Biol. Med., vol. 41, no.12, pp1110–1117.

[3] Saleha Khatun & Ruhi Mahajan (2016) "Comparative study of wavelet-based unsupervised ocular artifact removal Techniques for single-channel EEG data", IEEE Journal of Transitional Engineering in health and medicine, Vol. 4.Figure 5. Compare of EEG before and after denoising

[4] Zikov, T. & Bibian, S. (2002) "A wavelet based de-noising technique for ocular artifact correction of the electroencephalogram", Proceedings of the Second Joint EMBS/BMES Conference,pp98–105.

[5] Inuso, G. & Foresta, F.la. & Mammone, N. & Morabito, F. C.(2007) "Brain activity investigation by EEG processing: Wavelet analysis, kurtosis and Renyi's entropy for artifact detection", in Proc. IEEE Int. Conf. Inf. Acquisition, Seogwipo, South Korea, Jul, pp195-200.

[6] Mahajan, R. & Morshed, B. I. (2015) "Unsupervised eye blink artifact denoising of EEG data with modied multiscale sample entropy, kurtosis, and wavelet-ICA", IEEE J. Biomed. Health Informat, vol. 196, no. 1, pp158165, Jan.

[7] Yamaguchi, C, (2003) "Fourier and Wavelet Analyses of Normal and Epileptic Electroencephalogram (EEG)" in: Proc. of the 1st Intl. IEEE EMBS Conf. on Neural Engg, pp406–409.

[8] Akin, M,(2002) "Comparison of wavelet transform and FFT methods in the analysis of EEG signals", J. Med. Syst. 26 (3), pp241–247.

[9] Adeli, H. & Zhou, Z. & Dadmehr, N, (2003) "Analysis of EEG records in an epileptic patient using wavelet transform", Journal of Neuroscience Methods,123(1), pp69–87.

[10] Mallat, S. (1989) "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. Patt. Anal. Mach. Intell, vol. 11, pp674–693, July.

[11] Donoho, D. L(1993) "Nonlinear wavelet methods for recovering signals, images, and densities from indirect and noisy data", Proceedings of Symposia in Applied Mathematics, vol. 47, pp173–205.

[12] Krishnaveni, V. & Jayaraman, S. & Malmurugan N,(2004) "Non adaptive thresholding methods for correcting ocular artifacts in EEG", Academic Open Internet Journal, vol. 13.

[13] Tiwari ,A. & Khatwani P.(2013) “A survey on different noise removal techniques of EEG signals”, IJARRCE, vol .2, Issue 2, February, pp1091-1095.

[14] EEG time series data. https://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html

[15] Donoho, D.L,(1995) “De-noising by soft-thresholding”, Information Theory, IEEE Transactions ,Vol. 3,No. 41 pp613-627.

AUTHORS

Chen Ronghua, Graduate student of Tsinghua University. Research field is Biological signal filtering.

Li Dongmei, Associate professor of Tsinghua University. Research field is ADC and DAC design

Zhang Milin, Doctoral supervisor of Tsinghua University. Research field is BCI chip design and system research

INTENTIONAL BLANK

PERVCOMPRA-SE: A PERVASIVE COMPUTING REFERENCE ARCHITECTURE FROM A SOFTWARE ENGINEERING PERSPECTIVE

Osama M. Khaled, Hoda M. Hosny, and Mohamed Shalan

Department of Computer Science and Engineering,
The American University in Cairo, Cairo, Egypt

ABSTRACT

Pervasive Computing is a very challenging and complex domain that still lacks a comprehensive unified architecture. In this paper, we propose a reference architecture for pervasive computing that captures most, if not all, of the key challenges and provides a new architecture model that can be used in almost any business context. It provides conceptual views for the smart environment (SE), the smart object (SO), and the pervasive system (PS). We evaluated the model using a simulation prototype to predict its reliability at runtime.

KEYWORDS

Pervasive computing, Ubiquitous computing, Context-aware services, Internet of Things, Software architecture, Reference Architecture, Model-Driven Architecture, Software design

1. INTRODUCTION

Pervasive computing (PervComp) is one of the hottest topics for research nowadays.Its challenges exceed the outdated main frame and client-server computation models. Its systems are characterized as volatile, mobile, and resource-limited. They stream a lot of data from different sensors. In spite of these challenges, a PervComp system should be highly distributed and should receive multiple visits from different users using hybrid smart devices concurrently. Moreover, the system is expected to be sensitive to the environment and to adapt to the changes smartly and spontaneously.

The above implies by default, a lengthy list of quality features like context sensitivity, adaptable behavior, concurrency, service omnipresence, and invisibility. Such features entail additional challenges in the PervComp system like data security, privacy, quality of service, and fault tolerance. Consequently, these challenges require well-designed systems that consider all such features.

Fortunately, the device manufacturers improved their enabling technologies, such as sensors, network bandwidth, and batteries to pave the road for PSs with high capabilities. On the other

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 37–57, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80504

hand, this domain area has gained an enormous attention from researchers ever since it was introduced in the early 90s of the last century. Innovative systems have been applied in different business contexts such as learning, emergency, retail, and health. Albeit, it is still classified as one of the visionary systems that are expected to be woven into people's daily lives.

A unified architecture is one of the fundamental research challenges for PervComp systems [1] where a rapid and common architecture is much required. Ashraf and Khan [2] reported 26 challenges that are either not addressed or partially addressed. Some key architectural challenges, namely: Software Structuring, Integration, and conceptual modeling are among the top challenges that they found. Gazis et al. highlighted four architectural challenges in the Internet of Things (IoT) domain as well in a recent research paper surveying systems in the USA, Europe and China [3]. They named Reliability, Privacy and Security, interoperability, and device heterogeneity as the key challenges for the successful development of an IoT system.

The initiatives to provide a unified architecture are still very limited and focus on the IoT domain primarily[1]. It is worth mentioning here that there is already an existing RA for the IoT called IoT-A [4] since 2013; however, the IEEE Standards Association started another project to set architectural framework standards for the IoT domain. The project is active and has not been finalized until the writing of this document [5]. These initiatives focus mainly on the IoT, which mandates that objects should be Internet-enabled by definition, while PervComp, which is more generic, can accept objects to be Internet-enabled or not.

Moreover, the purpose of the unified architecture is not only to speed up the development process of a new software product, but more importantly is to bring all the software engineers into a common ground of understanding[6] by generating and sharing the same terminologies. Failing to interpret the different terminologies into common meanings can lead to a project's failure [4].

PervCompRA-SE provides a comprehensive reference architecture (RA) to generate concrete architectures for PervComp that can be adapted in different business contexts. It is a business-driven reference model covering 17 quality features with an extensive study in both the business and technical aspects of the RA. It resulted into practical business and technical models accompanied by guidelines and a trade-off analysis. We evaluated our technical model extensively using qualitative and quantitative methods.

The literature includes definitions for a Practice Reference Architecture (PRA) and a Futuristic Reference Architecture (FRA) [7].A PRA tries to capture best practices from existing architectures along with architectural patterns in order to facilitate the implementation of concrete architectures. Its intent is to resolve time-to-market and standardization problems. On the other hand, a FRA is built to become the first type. It must be based on research and it has to introduce innovative ideas [7]. Once an FRA is implemented as a concrete architecture it becomes an immature PRA, which encourages others to adopt it in more implementations and to transform it finally into a PRA.

----

[1] Some researchers label IoT as a branch from the PervComp and some others use the terminology to refer to the pervasive computing domain.

PervCompRA-SE is an FRA that captures best practices and introduces innovative features as well. The RA that we intend to build will be a visionary architecture. Hence, the focal point that this research addresses may be summarized as follows:

With the fast spread of pervasive systems, is it possible to generate a futuristic reference architecture for pervasive computing systems that encompasses most, if not all, architectural challenges and that can be applied/adapted in different business contexts?

The paper is organized as follows: section II provides our research methodology, section III covers the related RAs that we surveyed, section IV explains the technical reference architecture, section V describes the evaluation methods that we adopted, and section VI concludes the paper.

2. RESEARCH METHODOLOGY

PervCompRA-SE is designed to reflect the best practices in building reference architectures as well as capturing most, if not all, quality features in PervComp. The best practices approach that we adopted [6][8] is that the RA has to:

1. Capture the Essence of Existing Architectures.
2. Has an architectural baseline model.
3. Provides sufficient Guidance.
4. Considers the Business Needs.
5. Considers the Business Context.
6. Provides a Common Dictionary.
7. Captures and Shares Architectural Patterns.
8. Has an Architectural Vision.
9. Has a Prototype.

A PervComp system exhibits, as mentioned by Spínola[9], some key quality features. These are domain independent quality features that we classified into business and architectural quality features based on their proximity from the business and architectural contexts of PervComp.

We gave a clear description for the business quality features in our basic requirements model as explained in [10], which includes Adaptable Behavior (AB), Context Sensitivity (CS), Experience Capture (EC), Fault Tolerance (FT), Heterogeneity of Devices (HD), Invisibility (IN), Privacy and Trust (PT), Quality of Service (QoS), and Service Omnipresence (SO) [9]. We added Security (ST) and Safety (SY) to reach 11 business quality features.

We also adapted the 6 additional architectural quality features mentioned by Spínola[9], shown in Table 1.

We followed the normal software engineering lifecycle in order to collect the requirements, and generate the rest of the artifacts by exploring multiple sources in PervComp, and by using collected knowledge, and results of meetings with experts. We analyzed these requirements to generate additional artifacts (e.g. a business ontology, and quality features weights). We then moved to the next phase (design) in order to generate a technical reference architecture (TRA).

We generated the baseline architecture (BLA) model using the artifacts from the business analysis phase, and the artifacts generated from the design phase. Finally, we used different qualitative and quantitative techniques to evaluate the reference architecture (Fig. 1).

Table 1.Architectural Quality Features

Feature	**Description**
Concurrency (CON)	The system design must ensure proper performance and correct behavior of shared resources under concurrent access from different clients [9].
Function Composition (FCN)	The system must be able to produce new services from existing ones based on their specifications [9].
Openness (OPS)	It is a characteristic of a system which is measured by the number of key published services [9].
Scalability (SCL)	A system is scalable when it keeps operating, with an acceptable degree of efficiency, regardless of the increase in resources and users [9][11].
Service Discovery (SDV)	The system should be able to allocate new services, register them, and facilitate access to them according to the environment [9].
Spontaneous Interoperability (SIP)	The system should be able to associate itself with new partners (e.g. sensors, actuators, or peer systems) normally during operation [9].

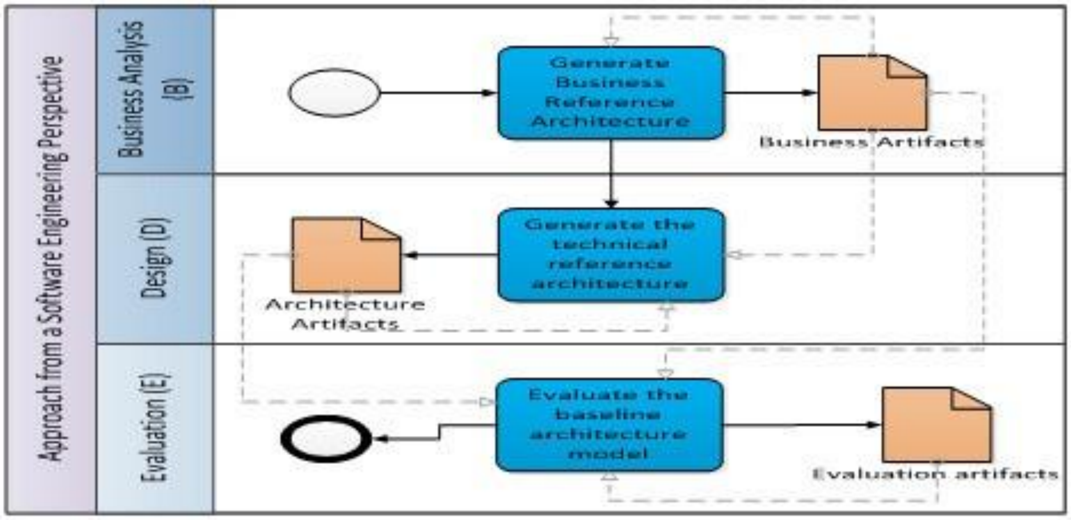

Approach from a Software Engineering Perspective
Business Analysis (B)
Design (D)
Evaluation (E)
Generate Business Reference Architecture
Business Artifacts
Generate the technical reference architecture
Architecture Artifacts
Evaluate the baseline architecture model
Evaluation artifacts

Figure 1. High-Level Approach from a Software Engineering Perspective

We organized the PervCompRA-SE so that the architect or the business analyst can use the business reference architecture (BRA) then proceed with the normal activities to generate a concrete architecture. On the other hand, the architect or the business analyst may proceed to review the TRA then proceed to generate the concrete architecture (Fig. 2). However, it is highly recommended to get acquainted with the concepts and terminologies in the BRA in order to generate a consistent and concrete architecture.

We derived the BLA model from the BRA [10], from the architectural requirements for the quality features shown in Table 1, from network challenges, technology enablers, and from relevant design and architectural patterns.

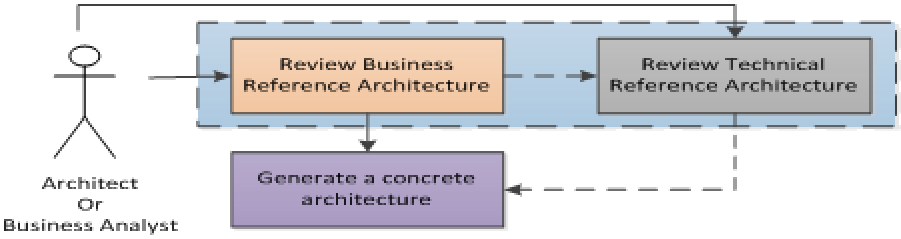

Review Business Reference Architecture
Review Technical Reference Architecture
Generate a concrete architecture
Architect Or Business Analyst

Figure 2. Decoupling BRA from TRA

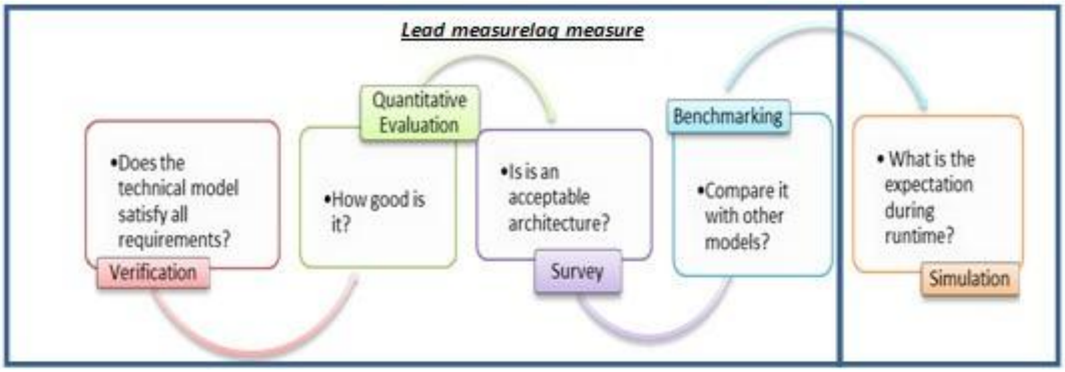

Lead measurelag measure
•Does the technical model satisfy all requirements?
Verification
Quantitative Evaluation
•How good is it?
•Is is an acceptable architecture?
Survey
Benchmarking
•Compare it with other models?
• What is the expectation during runtime?
Simulation

Figure 3. The suggested reference Architecture Evaluation Cycle

We adopted a hybrid approach that combined between qualitative and quantitative techniques. The evaluation cycle, as shown in Fig. 3, aims to trace every module in the BLA to the business and architectural requirements, generate measurements for the architecture metrics, compare these metrics with experts' models generated from the same set of requirements, and verify its acceptance by the development community.

The above evaluation activities give a *lead measure* of the design quality before implementation. They are sufficient for generating a concrete architecture. However, in order to predict its behavior during runtime, there is a need for a *lag measure*. We implemented a simulation experiment to predict the reliability of the PervComp system that adopts the PervCompRA-SE.

Although we provide a lot of abstracted concepts in PervCompRA-SE, we wanted to have special ontological terms that are derived from the quality features and give measurement scales that could be used at runtime. We captured ontological terms from the BRA and TRA and classified them either as *value* or as *issue*. The *value* is a benefit that system users need to gain from the system. The *issue* is a problem or a non-desired aspect that the system users are not willing to have [12].

3. RELATED WORK

There are very few research efforts which position themselves as RAs for the PervComp domain. Hence, we explored RAs from IoT as well because it comes very close to PervComp. We did not include early contributions in the PervComp domain which mainly focused on providing programming frameworks or middleware solutions (e.g. AURA, Gaia, SOCAM, CARISMA, CORTEX, and RCSM) [13][14] as they were irrelevant to our scope.

Table 2.Comparing Related Reference Architectures with Respect to quality features

RA FT	SO	IN	CS	AB	EC	SDV	FCN	SIP	HD	FT	ST	OPS	CON	QoS	SCL	PT	Score
(1)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓		✓		12
(2)	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓			✓	✓	12
(3)	✓		✓		✓	✓		✓	✓		✓					✓	8
(4)	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	15
(5)			✓	✓	✓					✓	✓			✓			6
(6)	✓	✓	✓	✓		✓	✓	✓	✓		✓		✓	✓	✓	✓	13
(7)			✓	✓						✓			✓	✓			5
(8)			✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	14
(9)	✓									✓		✓	✓	✓	✓		6
(10)			✓							✓	✓					✓	4
(11)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		15

We reviewed a number of related systems, namely: 1) I-Centric [15], 2) PCA_A [16], 3) Self-Care Infrastructures [17], 4) PSC-RM [18], 5) Smart Environment Software Reference Architecture [19], 6) the NGSON Multiplane Framework [20], 7) Component-based Self-Adaptive [21], 8) IoT-A [22], 9) CIPS [23], 10) IoT Security and Privacy [24], and 11) RA-Ubi[25]. Our revision aimed to speculate the level of satisfiability of the quality features and the abidance of the best practices approach as described in our approach (section II).

Most of the aforementioned RA's anchored on specific perspectives of PervComp architecture. For example, the Self-Care Infrastructures RA offered a RA suitable for a pervasive health environment. CIPS showed a RA for highly intensive data processing systems. NGSON highlighted a RA that network operators could adopt in order to provide PervComp solutions. A few of them tried to give generic views that could fit for any solution like RA-Ubi, IoT-A, PCA and I-Centric. Some others just focused on one architectural layer or component.

Tables 2 and 3 compare these RAs with respect to the quality features and the best practices approach, respectively as mentioned above, in section II.

We checked how many of the quality features (in Table 1), except safety, were fulfilled by these RAs as shown in Table 2. It is important to note that some RAs had a specific focus like the RA in (Security and Privacy in IoT) [21] which focused mainly on security and privacy. Other RAs focused on Environment Intelligence [19], and some others were oriented towards the pervasive services infrastructure [20]. The traced features as shown in Table 2 show the following:-

1. The IoT-A, PSC-RM and RA-Ubi considered most of the quality features essential for PervComp systems but the IoT Security and Privacy RA were the least to consider these features.
2. The quality features that the RAs considered most are context-Sensitivity followed by Service Security, adaptable behavior, and fault tolerance.
3. Function Composition and Openness were the least considered quality features.

We note from Table 3 that most of the RAs follow the best practices approaches. On the other hand the least adopted practices were: providing guidance to instantiate a new architecture or are based on a business context. Finally, all of them captured the essence of the existing architectures and were able to provide a common dictionary. This may simply mean that the authors were more concerned with explaining their concepts and making them clear for the readers.

The number of RAs focusing on PervComp is still limited and very few of them follow the best practices guidelines, as mentioned in [6] and [8], to build a robust RA and to cover most of the business challenges. Most of these RAs are not mature enough to provide enough guidance for software engineers and did not consider trade-offs among the quality features.

Table 3.Comparing Related Reference Architectures with Respect to Best Practices Approach

QC RA	Essence of Existing Architectures	Baseline Model	Guidance	Business Needs	Business Context	Dictionary	Patterns	Vision	Prototyping
(1)	✓			✓		✓	✓	✓	
(2)	✓	✓				✓	✓	✓	
(3)	✓	✓		✓	✓	✓			✓
(4)	✓	✓		✓	✓	✓	✓	✓	
(5)	✓		✓	✓	✓	✓			✓
(6)	✓	✓		✓		✓	✓	✓	✓
(7)	✓	✓	✓			✓	✓		✓
(8)	✓	✓	✓	✓	✓	✓	✓	✓	✓
(9)	✓	✓	✓	✓	✓	✓	✓		✓
(10)	✓			✓	✓	✓	✓	✓	✓
(11)	✓	✓		✓		✓	✓	✓	

4. THE TECHNICAL REFERENCE ARCHITECTURE

We The TRA is explored from technological, network, and design decisions [12] which resulted into a baseline architectural model describing the structure and behavior models as will be explained in the coming sections. The BLA model provides essential details about:

1. **The Smart Environment:** a conceptual view of the SE and classification of the objects.

2. **The Smart Object:** an abstracted view of the SO and the essential handlers that it should include to interact with the SE.

3. **The Pervasive System:** The essential modules that should exist in a PS with high level linkage among them.

4. **The System Optimization:** The basic optimization parameters in the system.

5. **The Architecture Variability:** the essential configurations of the PervCompRA-SE to generate different architectural models based on the changing rules.

6. **The System Deployment:**The essential deployment strategies that could be implemented for a PS in order to increase its reliability.

4.1 Smart Environment

The SE is just an instantiation of the PervComp system where objects show a high degree of intelligence. Ideal SOs possess processing powers (memory & processor), a communication interface, sensors and actuators. According to Kortuem et al. the degrees of smartness could be there among objects based on the manufacturers' designs. Such degrees are categorized into three types [26]. Each type has its associated set of functions, rules, and workflows:

1. **Activity-aware object**: this is an object that can record information about the surrounding activities and aggregates them, but does not respond to these activities.

2. **Policy-aware object**: this is an object that can recognize surrounding activities according to pre-defined policies and devises proper actions and hence can respond by a warning or an alert.

3. **Process–aware object**: this is an object that recognizes surrounding activities in the light of organizational processes and provides proper directions for users about tasks, deadlines, and decisions.

It is important to note here that the SE can be composed of other passive objects that are not smart by design such as RFID-tagged devices which can be identified only by other sensor-enabled objects, which could be SOs as defined earlier. For example, tracking boxes of products coming in/out of a specific warehouse does not require intelligence in these boxes. They just need a reader and RFID stamp-tags per box.

Hence, we reached a generic model for the SE, which is ideally represented as a PS as shown in Fig. 4. The SE is structured as follows:

1. The SE can have a nested SE. Every SE is composed of objects.

2. An object could be a SO or a dummy object. The details of the SO are derived from Microcontrollers and Smart Phones.

3. A SO is classified as Activity-Aware, Policy-Aware, or Process-Aware. It can contain

dummy objects.

4. A dummy object is an object that lacks one of the properties of the SO. It has a specific job responsibility with no intelligence or logic. A dummy object is either an active object or a passive object as explained above.

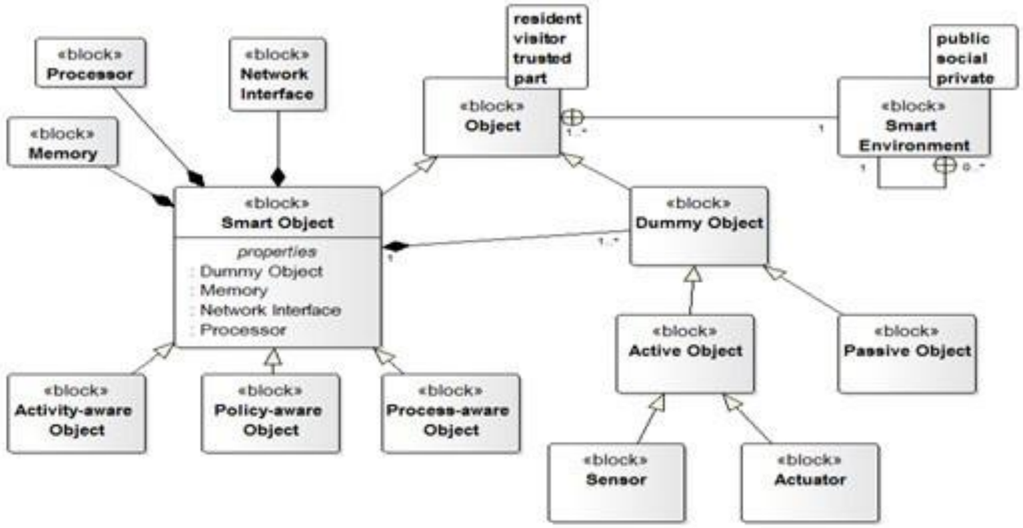


Figure 4. Pervasive Computing Analysis Approach

As shown in Fig. 4 also, a SO must possess some properties, or capabilities, namely processor, memory, network interface, and some sensing or actuation capabilities. We defined some types for the object and the SE, which helps the architect to take better decisions. The SE is an environment that exhibits intelligence behavior through SO(s) that are part object(s) or resident object(s). The SE can be classified from a privacy point of view into [27]:

1. **Public**: where most of its services and resources are accessible to its objects with no access rules.

2. **Social**: that is an environment that grants access to its resources and services based on group association.

3. **Private**: the resources and services are accessible to objects that have the proper permissions for them only.

An object is anything in the world which can be represented in the SE. A classification of the objects based on their interaction model with the SE could be as follows:

1. **Part Object**: an object which cannot be removed from the system, else the system will not function as designed.

2. **Resident Object**: an object which is important as it accomplishes one or more tasks of the system, but removing it will not hinder the system design.

3. **Trusted Object**: an object that the system trusts and that joins the environment frequently.

4. **Visitor Object**: a non-trusted object that joins the environment in ad-hoc situations

All types of objects that join the SE need to interact with the environment in the most optimum way. Hence, there are two types of configuration approaches that can be adopted [11]:

1. **Preconfigured**: the object is bound to the environment through a configuration that aims to establish a long-term relationship between the object and the environment. It applies mostly to the part objects and may be applied to the resident objects.

2. **Spontaneous**: the object is bound to the environment through a spontaneous configuration. This type of configuration applies to the visitor, resident, and trusted objects. The spontaneous binding requires from the system that it negotiates first with the device using a standard protocol, then the system binds it, then the object starts interaction using the proper protocol which was agreed upon during the negotiation step.

4.2 Smart Object

The SO is an important part of a successful PS. It can be programmed to provide the required behavior and can carry out different roles in the SE. Accordingly, we recommend standardizing the SO with handlers that can address key issues. These handlers can add more controls on the PS.

The developer needs not only know how to program the SO, in case its interface is available for any programmer, but also needs to know extra details that are considered essential for robust and safe PSs. Moreover, the final architecture mainly depends on the capabilities of the devices that compose the skeleton of the system. Some usage scenarios of SOs may put some living creatures' lives at risk [28]. Hence, we recommend the following standards for SOs, which we described in details in [29]:

1. **Programming Permissions:** objects in a physical world, may risk lives if not used properly, as well as impact privacy and security of people. The object should hence provide three protection levels for its programmable interface :
    a. **Public:** the API is accessible for the designers without permission from the manufacturer.
    b. **Protected:** the API is accessible for the certified designers by the manufacturer.
    c. **Private:** the API is accessible only by the manufacturer's engineers.
2. **Safety procedures:** the SO must supply all safety procedures either as APIs, configuration, or documentation to read in order to avoid risking the context in which it runs.
3. **Security and privacy procedures:** the SO should provide sufficient APIs that guarantee data security and the protection of the confidential data.

4. **Volatility status:** the SO should provide APIs to determine its volatility status and help in predicting its disappearance from the environment.
5. **Processing Power status:** the SO should provide APIs to reveal its processing availability and memory status.
6. **Process Hosting:** A SO should cooperate with its environment and provide hosting capabilities to execute tasks if there is room for it (e.g. CPU processing is 1% and there is enough memory and storage).
7. **Community statistics:** the SO should provide APIs to gather statistics about it in order to share with the development communities. APIs should not reveal any confidential data. It will help software engineers understand how to deal with different SOs in different environments.

The SO can run in different modes:

1. **Runtime**: where all handlers run with full capacity and with minimal overhead.
2. **Diagnostics**: the SO adds extra overhead to its handlers, like logging, memory dump, etc.
3. **Maintenance**: the SO is in maintenance mode, which means that some of its functions may not be available. For example, its network interface may be disabled, or the handlers that will be disabled will notify the callers that it is in maintenance mode.

4.3 Pervasive System

The PS's behavior is centered on a basic workflow (Fig. 5), by which input devices feed in the system with an event and the system gathers events as context, interprets the context, and then links the interpretation to a decision. The decision, may or may not, trigger actions. These actions are made with output devices, which in turn feedback the system with their results as input data. The workflow cycle is inspired from the human perception process described in [10].

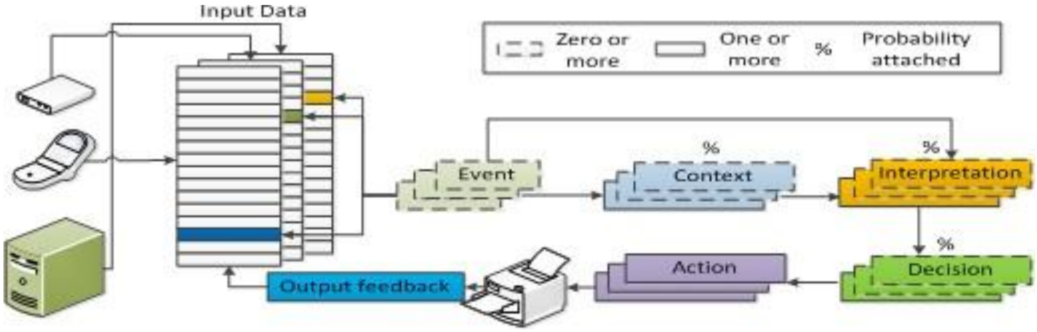


Figure 5. Basic operation workflow in a pervasive system

The PS should consider the uncertainty of the context, interpretation, and decision since the event could lead into different contexts, different interpretations, and different decisions. The basic behavior shows important concepts to understand the whole workflow.

The cycle starts with an **Input** which is a device capable of sending data to the system. The input devices can be classified into **Explicit** and **Implicit** inputs**. Explicit** is an input device that feeds the system with external data and requires direct interaction with the system (e.g. keyboard and mouse). **An implicit** input device is a device that feeds the system with external data by detecting the data from the environment (e.g. sensors). The sensor could be a physical or a virtual sensor. It could be a dummy object or a part object in a SO.

A virtual sensor is a software sensor that reads data from other software systems. An example of a virtual sensor is the social network sensor, which reads the status of the user all the time and sends it as an input to the system. A **physical sensor** is a physical device that reads environment conditions like heat, pressure, and light sensors.

The system fetches an **event** which is the basic incident that stimulates the system. The event is identified based on sensed data from different physical and virtual sensors. After that the system identifies a **context** whichis a specific status of the system identified by a set of parameters, a sequence of one or more events. The event can give an indication for one or more contexts; each one may have a different *occurrence weight*. It is described as c = $(e_1, e_2, ..., e_n)$ as the system determines the context by detecting a finite sequence of events from 1to *n*. It is important to note that actions of the system may trigger new events that subsequently may lead to new changes in the context.

Interpretation is the logical meaning of the context**.** One or more contexts could have the same interpretation, or the context could have more than one possible interpretation. The events and the context determine the right interpretation. Different interpretations could have different weights as well. The **interpretation** may lead to a **decision** that can be taken. There could be more than one decision, each one with a different weight. The **decision** can lead to zero or more **actions** which are the results of the system **decision.**

The action can be either **visible** or **invisible.** An **invisible** action is taken by the system within its components and does not require direct attention from the users. For example, a self-maintenance action to reallocate resources or free memory is considered an invisible action. The user may review this action later on from the system logs. A **visible** action requires direct attention from the user. For example, a warning message displayed on a screen is considered a visible action. Opening a door as the user steps forward is considered a visible action. A visible action can be further classified as **silent** and **interactive**.

A **silent** action does not require a reaction from the user, e.g. the message or video displayed on a screen. An **interactive** action requires a reaction from the user. For example, switching the light due to opening the room door is an interactive action. Acknowledgment of receiving a warning message is an interactive action.

The system takes actions through an **output** which is a mechanism of the system to make actions. It sends its **output feedback** as the result of the output device, back to the system as an input data.

The whole system may run in different modes. A mode is a special status of the system where operations may have different inputs, execution scenarios, and different outputs. However, modes, in general, will run the system as in its basic operations. The PS should have the following basic modes:

1. **Runtime:** the system runs all its operations in the optimum way.
2. **Assertion:** this is an administrative mode, where details of the system activities are revealed only to the administrator and logged for further analysis.

3. **Out of Service:** this mode should be used if the system should not be shut down and at the same time receives requests but without processing them.
4. **Upgrade:** the system is under upgrade operation which makes one or more modules unavailable until the upgrade process is complete.

Advanced modes could be added to the system to test the results of specific inputs, and outputs or to teach the system and let it improve its rules:

5. **Simulation:** this mode imitates the real world by running hypothetical scenarios over time as if it is running in the real world [30][31].

6. **Teaching:** The system will be in this mode if a lot of details are required in order to feedback the system to improve its artificial intelligence rules [30]

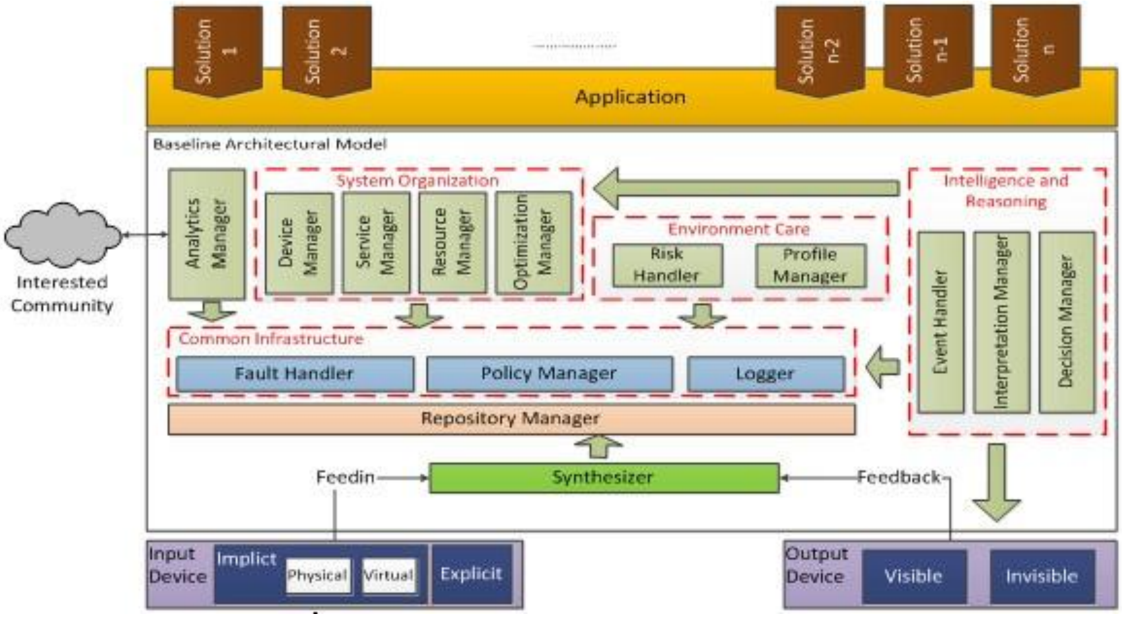


Figure 6. Pervasive System baseline architecture

Fig. 6 shows the structure of the baseline architectural model that we propose for a PervComp domain.

Externally from the architectural view, there is an **application** that implements the functional and quality requirements of the system. Some of these functional requirements are implemented as *solutions*. There could be one or more **solutions** for specific problems installed in the system as plugins. They can be installed/uninstalled in a systematic way. The solution may interact with the core modules of the system.

The **Repository Manager** is the place where data, information, knowledge, and wisdom are stored. The *Repository Manager* is responsible for coordinating the repository operations. The **Synthesizer,** on the other hand is responsible for receiving input from input devices and feedback from output devices, validating them, correcting them if required, and then saving them in the repository. The *Synthesizer* can be part of a middleware and it could also be a built-in service provided by the manufacturer of the input or output devices.

There are **common infrastructure** layers that serve most of the system modules. It is recommended to interact with them *asynchronously* in order not to impact the overall performance:

1) The **Logger** is used to log events, capture system performance in log files which could be in different formats.

2) **Policy Manager:** The policy manager is responsible for managing the system policies and the pre-defined configuration parameters.

3) The **Fault Handler** is responsible for handling all types of faults and taking the proper actions based on the system design as described in [10]. The *Fault Handler* cooperates with the *Interpretation Manager* and the *Decision Manager* to improve its performance.

The **intelligence and reasoning** layer, as shown in Fig. 5, is responsible for handling the basic workflow. Its main components are:

1) **An Event Handler:** is responsible for detecting the events and transforming them into contexts, or linking them with decisions. The *Event Handler* can be part of a **middleware**. The system can assign all middleware responsibilities to it since it is the main interaction point with the rest of the modules. The *Event Handler* may delegate the event to one of the system modules to handle or ignore it if it is already defined to be handled by other modules.

2) **An Interpretation Manager:** The *Interpretation Manager* is responsible for analyzing the interpretation rules and finding new correlations leading it to enhance its reasoning.

3) **A Decision Manager:** The D*ecision Manager* is responsible for analyzing the set of decision rules which are combinations between contexts and decisions with specific weights.

The **system organization** layer is responsible for managing system devices, services, and resources:

1) The **Device Manager** is responsible for registering and tracking all the devices that interact with the system with enough details like (device name, version, manufacturer, manufacturing date, OS version, binding date, last interaction date, unbinding date, display dimension, battery lifetime, etc ...).

2) The **Resource Manager** is responsible for registering the system resources, their locations, their availability per time unit, and their allocation with other objects.

3) The **Service Manager**, which can be part of a middleware, is responsible for registering new services, binding, unbinding, handing over services for mobile users, and producing new composite services.

4) The **Optimization Manager** is responsible for optimizing different system components for the best utilization of services and resources. It is concerned with optimization for quality attributes like processing time, availability, scalability, and responsiveness with respect to the functionalities of the system.

The **Environment Care** layer contains the modules that manage the profiles of the users and risk issues:

1) The **Profile Manager** is responsible for maintaining users' profiles including their preferences and tracking their activities and recording their behavioral trends.

2) The **Risk Handler** is responsible for handling, analyzing, and taking counter measure actions regarding all events concerning highly protected zones (Security, Safety, and Privacy & Trust issues).

The remaining module, the **Analytics Manager,** is responsible for preparing the required statistics about the system. For example, it can aggregate data collected by the *Logger* to show the system performance with different static quality features, like *Context Sensitivity* or *Scalability*, or runtime quality features, e.g. network throughput or reliability. It is responsible also for generating information and knowledge about the system. Some of these statistics will be shared with the interested communities through services published by the *Service Manager*. The **interested Community** is a cloud or a system with details about the usage of devices in different environments.

4.4 System Optimization

The PS's behavior is in continuous change. The behavior may not be correct all the time and accordingly, it needs to be continuously optimized. The optimization process is conducted to assign proper weights for different factors inside the system with predefined standard deviation for every factor. These factors control the behavior of the system and make its choices more accurate. It can be described as a function of a 5-tuple *(Q, C, I, D, S)*:

- $Q = \{q_i \mid i = 1,2,\ldots,n\}$ is a finite set of weights for quality features.
- $C = \{c_i \mid i = 1,2,\ldots,m\}$ is a finite set of weights for different contexts that the system may be in at any point of time
- $\mathrm{I} = \{i_j \mid j = 1,2,\ldots,r\}$ is a finite set of weights for different interpretations that the system may use to interpret its current situation.
- $\mathrm{D} = \{\mathrm{d}_i \mid i = 1,2,\ldots,k\}$ is a finite set of weights for different decisions that the system may take to adapt itself to the change in the context.
- $\mathrm{S} = \{\mathrm{s}_i \mid i = 1,2,\ldots,h\}$ is a finite set of weights for different solutions that the system may use to implement the adaptation decision.

4.5 Architecture Variability

The *Policy Manager*is responsible for enforcing guiding behaviors on the system. It encompasses the variable behavior of the system during runtime. It is possible to provide different preplanned settings for the *Policy Manager*. The setting may enable/disable some components or features in the system to provide a required behavior. This is not an *Adaptable behavior*, as the adaptability of system will work within the guidelines of the selected policy. We can have what is called a *Dynamic Architecture* of the system. The *Dynamic Architecture* may be defined by the configuration settings in the *Policy Manager.* All these settings manage the component behavior or its relationships with other components [32]:

1. **Enable/Disable Solutions:** solutions could be installed in the system as plugins. There could be different plugins providing similar services but with different functionalities. The *Policy Manager* may choose a policy file with a specific set of solutions based on the mode and the context.

2. **Roles and Responsibilities:** a policy may define a different role and responsibility for a specific object, which is ideally a smart device or a server.

3. **Service product line workflow:** organizes the services or functions, as requested by the *Compose Functions* architectural quality feature.

An architect willing to produce an architecture in a *Product Line Architecture* model may follow the following approach:

- The architect may set the weight for every quality feature or use the default ones as explained in [10].
- Design all possible solutions to resolve the functional and quality problems.
- Set a weighted score for the solutions based on their positive and negative impacts on quality features as shown in our work [33].
- Choose the solutions with the highest scores to produce the design.

The *Product Line Architecture* may change the weights of the quality features and subsequently the solution weights may change, which may lead to a different architecture model.

4.6 System Deployment

There are 3 basic roles that any module/component in the system should be able to assume:

- **Client**: the component requests services from other components.
- **Server**: the component offers services for other components.
- **Peer**: the component requests and offers services.

The classification of the objects in the SE, as shown in Fig. 4 imposes standard preference. An object which is part of the PS should ideally cooperate with other objects in the system to provide the required services for their clients which are usually trusted or visitor objects. However, the system may include some resident objects that can offer services as well, although they can request services from the system as clients. The trusted or visitor objects can act as peers and request/offer services from/to each other if the system is P2P. The sensor offers services for the system since it collects data from the environment. Clients can pull sensor’s data upon need. The actuator is similar to the sensor, but it offers actions. In summary, the level of responsibility of the object to produce or consume a service controls the role of the object as client, server, or peer as shown in Table 4.

Table 4.Objects in a Smart Environment and Expected Roles

	Part	**Resident**	**Trusted**	**Visitor**	**Sensor**	**Actuator**
Client	Low	Med	High	High	Low	Low
Server	High	High	Low	Low	High	High
Peer	Med	Med	High	High	Low	Low

5. EVALUATION

As PSs are considered complex, they cannot follow the traditional development cycle. It is recommended to test them first using simulation approaches [34]. We designed a discrete-event simulation experiment in order to predict the reliability of the BLA at runtime. A reliable system is a system that can perform its assigned functionality with a high probability of success during a specified period of time and within specific design constraints [35]. It increases the confidence in the design and gives an early indication for the expected behavior of the system.

We implemented a scenario for tracking a bus equipped with speed and location sensors that travels from point A to point B. The system receives visits from users and optimizes its modules using shared resources as it tracks the trip. The system runs in different modes as well.

We devised a conceptual model towards the complete implementation. Our model is derived from the PervCompRA-SE smart environment conceptual model as described above, in section IV. The model is composed of Entities classified as part objects, resident objects, and visitor objects. The part objects are those modules that define the baseline architecture as shown in fig. **6.** The sensors and actuators are resident objects that the part modules interact with to receive data and output data. The smart objects are visitors that request services from the system on regular basis (fig. 7). *Entities* interact with each other through their input and output ports. Every simulation module has two basic attributes *phase* and *tick*. *Phase* represents the status of the entity and *tick* is the logical time by which the entity can accept inputs and generate outputs.

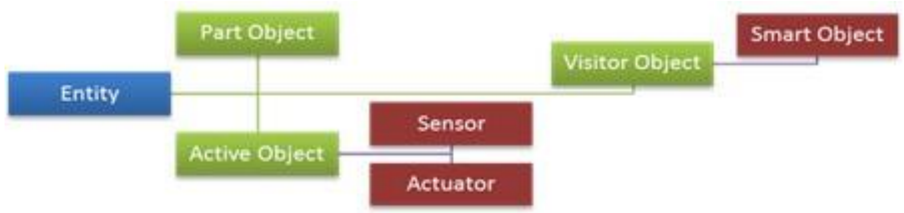


Figure 7. The Simulation Conceptual Model

The whole simulation model can be working in Runtime, Assertion, Security Threat, or Out of Service modes. First, the Runtime mode is the normal execution scenario without changes in the settings of the entities. Second, the Assertion mode is the normal execution scenario but with additional logging activities from these entities. Third, the Security Threat mode is where the whole system is threatened and needs to take some measurements to protect itself. It rejects visits from new smart objects recognized as visitors and accepts visits from the trusted smart objects only. It disables the Synthesizers so that no data can be collected from the sensors. Finally, the Out of Service mode is where the system will not be processing sensor signals, will not accept visits, and will not fetch user profiles from the Repository Manager. The system will still keep recording sensor data and when the system returns to one of the other three modes, then it can fetch the data and continue processing.

Pervasive systems, or IoT systems, are highly vulnerable to security threats [36]. Moreover, systems usually go out of service due to planned maintenance or unplanned outages. It is noticed that the cost of maintaining a system is in continuous increase since the end of the last century. This is basically due to the increased number of developed software applications and their increased complexity [37]. Moreover, administrators dump logs from the system for monitoring purposes all the time. Accordingly, we assumed that the system will be running in normal mode most of the time (64-70%) and there is 30-36% probability that it will be running in one of the other abnormal modes (Assertion, Out of Service, or Security Threat).

We derived other probabilities from different sources to prepare our conceptual model for the simulation with respect to failures of the speed and location sensors, hardware failures, SMS engine reliability, digital screen failure, battery recharge threshold, and rate of accidents. We also assumed that the complexity degree of the module (part object), as shown in equation 1, impacts the probability of failure as well as the probability of repair [38][39].

$$\mathrm{weig}h\mathrm{t} = \mathrm{Round}\left(\frac{\mathrm{r} * \mathrm{d}}{\sum_{\mathrm{i}=0}^{15} \mathrm{r_i} * \mathrm{d_i}} * 100\right) \qquad (1)$$

Where *r* is the number of satisfied requirements by the part object and *d* is the number of input and output dependency relationships for the part object.

We devised 6 different scenarios with different configurations and different control variables. The configurations were related to the fault handling, optimization, and execution modes. The control variables were divided into best, average, and worse values. We implemented the simulation project using DEV-Suite [40] and Java 6. We executed a total of 45 runs in order to increase the accuracy of prediction.

We captured the Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR) [41] as per equation 2. A reliability measurement is a function in MTBF and gives a score between 0 and 1 [39].

$$\mathrm{Reliability} = \frac{\mathrm{MTBF}}{\mathrm{MTBF+1}} \mathrm{Availability} = \frac{\mathrm{MTBF}}{\mathrm{MTBF+MTTR}} \qquad (2)$$

The experimental results showed that the model has in the worst cases a reliability of 96.86% and availability of 90.89%. In the average cases, the reliability of the system is 98.08% and availability is 95.77%. In the best cases the reliability is 99.9% and availability is 99.79%.

We also predict an average of 2% additional time needed from the last sensor input calculated against the perfect scenario. The results show that the resource optimization technique that we adopted is working reasonably as it saved 3% of the potential failures of the modules. It was evident that the assumptions for the control variables dominated the general performance of the systems. In general, the scenarios show that the processing time increases as the working conditions get worse.

Although the analysis shows positive results about the reliability and the availability of the architecture model, the prediction is an initial estimation which will definitely change in a real environment. There should be a continuous improvement process by fetching real numbers about the systems' performance during runtime in order to give more accurate predictions about the system failure.

6. CONCLUSION

The pervasive domain is a very complex domain and by providing a unified architecture, standard and robust systems can be more easily implemented. We provided our vision for a reference architecture in the PervComp domain. This vision is based on a detailed study of the business requirements of the PervComp domain as well as the architectural challenges that may be encountered. We provided the details of the simulation prototype which we used to predict the quality of the PervCompRA-SE.

The PervCompRA-SE is a business-driven, safety-aware, open, simple, and almost complete reference architecture. Its design is derived from an extensive study of different business areas and quality features for the PervComp domain. It is one of the few RAs in this domain to handle safety issues. It is not a complex model and its concepts are easy to understand. It addresses 17 quality features through a process that abides by the best practices which makes it offer one of the best choices in this domain for implementation.

We plan to report about the remaining contributions and give more details about the architectural challenges, about the evaluation approaches, and other details of the TRA. It is quite a challenging research area and we plan to continue our work in software product line driven from our reference architecture. We aim to have a large-scale implementation using our architectural reference model as well.

REFERENCES

[1] W. Dargie, J. Plosila, and V. De Florio, "Existing Challenges and New Opportunities in Context-aware Systems," in Proceedings of the 2012 ACM Conference on Ubiquitous Computing, New York, NY, USA, 2012, pp. 749–751.

[2] M. U. Ashraf and N. A. Khan, "Software Engineering Challenges for Ubiquitous Computing in Various Applications," in 2013 11th International Conference on Frontiers of Information Technology, 2013, pp. 78–82.

[3] V. Gaziset al., "Short Paper: IoT: Challenges, projects, architectures," in 2015 18th International Conference on Intelligence in Next Generation Networks, 2015, pp. 145–147.

[4] "Internet of Things - Architecture IoT-A. Deliverable D1.5 – Final architecture reference model for the IoT v3.0." European Lighthouse Integrated Project, Jul-2013.

[5] IEEE Standards Association, "IEEE Project (P2413) - Standard for an Architectural Framework for the Internet of Things (IoT)," Dec-2016. [Online]. Available: http://standards.ieee.org/develop/project/2413.html. [Accessed: 21-Apr-2017].

[6] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, and M. Bone, "The Concept of Reference Architectures," Syst. Eng., vol. 13, no. 1, pp. 14–27, 2010.

[7] S. Angelov, J. J. M. Trienekens, and P. Grefen, "Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case," in Software Architecture: Second European Conference, ECSA 2008 Paphos, Cyprus, September 29-October 1, 2008 Proceedings, R. Morrison, D. Balasubramaniam, and K. Falkner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 225–240.

[8] E. Y. Nakagawa, "Reference Architectures and Variability: Current Status and Future Perspectives," in Proceedings of the WICSA/ECSA 2012 Companion Volume, New York, NY, USA, 2012, pp. 159–162.

[9] R. O. Spínola and G. H. Travassos, "Towards a framework to characterize ubiquitous software projects," Inf. Softw. Technol., vol. 54, no. 7, pp. 759–785, 2012.

[10] O. M. Khaled, H. M. Hosny, and M. Shalan, "A Pervasive Computing Business Reference Architecture: The Basic Requirements Model," Int. J. Softw. Eng. IJSE, vol. 10, no. 1, pp. 17–46, Jan. 2017.

[11] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, Distributed Systems: Concepts and Design, 5th ed. USA: Addison-Wesley Publishing Company, 2011.

[12] O. M. Khaled, "Pervasive Computing Reference Architecture from a Software Engineering Perspective," Ph.D. Dissertation, The American University in Cairo, Cairo, Egypt, 2017.

[13] H. Vahdat-Nejad, "Context-Aware Middleware: A Review," in Context in Computing, New York, 2014, pp. 83–96.

[14] D. Romero, "Context-Aware Middleware: An overview," Paradig. Oscar Gonzalez, vol. 2, no. 3, pp. 1–11, 2008.

[15] R. Popescu-Zeletin, S. Steglich, and S. Arbanowski, "Pervasive communication: a human-centered service architecture," in Proceedings. 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, 2004. FTDCS 2004., 2004, pp. 140–146.

[16] Y. Liu and F. Li, "PCA: A Reference Architecture for Pervasive Computing," in 2006 First International Symposium on Pervasive Computing and Applications, 2006, pp. 99–103.

[17] G. Roussos and A. Marsh, "A blueprint for pervasive self-care infrastructures," in Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006, p. 6 pp.-pp.484.

[18] J. Zhou et al., "PSC-RM: Reference Model for Pervasive Service Composition," in 2009 Fourth International Conference on Frontier of Computer Science and Technology, 2009, pp. 705–709.

[19] A. Fernandez-Montes, J. A. Ortega, J. A. Alvarez, and L. Gonzalez-Abril, "Smart Environment Software Reference Architecture," in 2009 Fifth International Joint Conference on INC, IMS and IDC, 2009, pp. 397–403.

[20] J. Liao, J. Wang, B. Wu, and W. Wu, "Toward a multiplane framework of NGSON: a required guideline to achieve pervasive services and efficient resource utilization," IEEE Commun. Mag., vol. 50, no. 1, pp. 90–97, Jan. 2012.

[21] L. C. Bueno, "A Reference Architecture for Component-Based Self-Adaptive Software Systems," Department of Information and Communication Technologies Facultyof Engineering, ICESI University, Cali, Columbia, 2012.

[22] "Internet of Things Architecture IoT-A Project Deliverable D6.2 – Updated Requirements." European Lighthouse Integrated Project, Jan-2011.

[23] R. Al Ali, I. Gerostathopoulos, I. Gonzalez-Herrera, A. Juan-Verdejo, M. Kit, and B. Surajbali, "An Architecture-Based Approach for Compute-Intensive Pervasive Systems in Dynamic Environments," in Proceedings of the 2Nd International Workshop on Hot Topics in Cloud Service Scalability, New York, NY, USA, 2014, p. 3:1–3:6.

[24] I. D. Addo, S. I. Ahamed, S. S. Yau, and A. Buduru, "A Reference Architecture for Improving Security and Privacy in Internet of Things Applications," in 2014 IEEE International Conference on Mobile Services, 2014, pp. 108–115.

[25] C. A. Machado, E. Silva, T. Batista, J. Leite, and E. Nakagawa, "RA-Ubi: A Reference Architecture for Ubiquitous Computing," in Software Architecture: 8th European Conference, ECSA 2014, Vienna, Austria, August 25-29, 2014. Proceedings, P. Avgeriou and U. Zdun, Eds. Cham: Springer International Publishing, 2014, pp. 98–105.

[26] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the Internet of things," IEEE Internet Comput., vol. 14, no. 1, pp. 44–51, Jan. 2010.

[27] V. Kostakos, E. O'Neill, and A. Penn, "Designing Urban Pervasive Systems," Computer, vol. 39, no. 9, pp. 52–59, Sep. 2006.

[28] H. I. Yang and A. Helal, "Safety Enhancing Mechanisms for Pervasive Computing Systems in Intelligent Environments," in 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), 2008, pp. 525–530.

[29] O. M. Khaled, H. M. Hosny, and M. Shalan, "On the road to a reference architecture for pervasive computing," in 2015 International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS), 2015, pp. 98–103.

[30] M. Wollschlaeger, S. Theurich, A. Winter, F. Lubnau, and C. Paulitsch, "A reference architecture for condition monitoring," in 2015 IEEE World Conference on Factory Communication Systems (WFCS), 2015, pp. 1–8.

[31] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, Discrete-Event System Simulation, 5th ed. Prentice Hall, 2010.

[32] E. Cavalcante, T. Batista, and F. Oquendo, "Supporting Dynamic Software Architectures: From Architectural Description to Implementation," in 2015 12th Working IEEE/IFIP Conference on Software Architecture, 2015, pp. 31–40.

[33] O. M. Khaled, H. M. Hosny, and M. Shalan, "A Statistical Approach to resolve conflicting requirements in pervasive computing systems," presented at the The 12th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2017), Porto, Purtogal, 2017, p. 12.

[34] S. S. Brink, "Enabling Architecture Validation in the Analysis Phase of Developing Enterprise or Complex Systems using Enterprise Architecture Simulation Environment (EASE)," in MILCOM 2007 - IEEE Military Communications Conference, 2007, pp. 1–8.

[35] R. Roshandel, N. Medvidovic, and L. Golubchik, "A Bayesian Model for Predicting Reliability of Software Systems at the Architectural Level," in Proceedings of the Quality of Software Architectures 3rd International Conference on Software Architectures, Components, and Applications, Berlin, Heidelberg, 2007, pp. 108–126.

[36] D. Storm, "Of 10 IoT-connected home security systems tested, 100% are full of security FAIL," COMPUTERWORLD, 11-Feb-2015.

[37] J. De Vries, C. Burki, and B. De Vries, "How to save on software maintenance costs," Omnext, Nov. 2014.

[38] Y. Liu and I. Traore, “Complexity Measures for Secure Service-Oriented Software Architectures,” in Predictor Models in Software Engineering, 2007. PROMISE’07: ICSE Workshops 2007. International Workshop on, 2007, pp. 11–11.

[39] J. Iqbal, D. S.M.K.Quadri, and T. Rasool, “On Way to Acquiring Reliability Growth in Software Systems,” Int. J. Comput. Appl., vol. 24, no. 7, pp. 33–36, Jun. 2011.

[40] “DEVS-Suite,” Arizona Center for Integrative Modeling& Simulation, 2.0.

[41] H. Pham, System Software Reliability. Springer London, 2006.

AUTHORS

Osama M. Khaled received his Bachelor, Masters and PhD degrees in Computer Science from the American University in Cairo in 1998, 2004, and 2017 respectively. Osama works in the software development and telecommunication industry since 1998 and acts as a lecturer and consultant in software engineering as well. His active research areas are in software engineering, software architecture, software modelling, business analysis, and software programming. Osama is the author/co-author of 12 publications in international journals and conference proceedings.

Hoda M. Hosny is a Professor of Software Engineering and has been teaching at the American University in Cairo since 1985. She started her teaching career in the University of California, Davis (UCD) in 1981. She received her Masters degree in Computer Science from UCD in 1984 and her Ph.D. from Leeds University, UK, in 1991. She served as an IT Specialist, Consultant and Trainer at a number of National and International Institutions and was invited to lecture at 4 other universities in Cairo. She is the author/co-author of more than 50 publications in international journals and conference proceedings.

Mohamed Shalan is an Associate Professor (with tenure) at the Department of Computer Science and Engineering, the American University in Cairo. He received his Ph.D. in computer engineering from the Georgia Institute of Technology (GaTech) in 2003. He received his B.Sc. and M.Sc. in computer and systems engineering from Ain Shams University, Cairo, Egypt in 1993 and 1997. His research interests are in the area of computer engineering, with focus on embedded systems, digital design, energy-efficient computing systems, and electronic design automation. Professor Shalan has over 30 refereed conference and journal papers. Also, he holds 2 US patents.

INTENTIONAL BLANK

A SIMULATION APPROACH TO PREDICATE THE RELIABILITY OF A PERVASIVE SOFTWARE SYSTEM

Osama M. Khaled, Hoda M. Hosny and Mohamed Shalan

Department of Computer Science and Engineering,
The American University in Cairo, Cairo, Egypt

ABSTRACT

The pervasive computing domain is a very challenging one and requires a robust architectural model to facilitate the production of its systems. In this paper, we explain a case study using a simulation prototype to validate our baseline architecture of a reference architecture for the pervasive computing domain. The simulation prototype was very useful in predicting the reliability and availability of the system using the baseline architecture during runtime.

KEYWORDS

Pervasive Computing, Ubiquitous Computing, Software Engineering, Software Architecture, Software Validation and Verification, Internet of Things, IoT, Discrete Event Simulation

1. INTRODUCTION

Software Architecture is one of the critical tasks in the software development lifecycle. The design decisions that the software architect takes represent the skeleton of the software system. Incorrect decisions may lead to a complete failure of the whole system. Moreover, correcting the wrong designs could be very expensive for both the development team and the customer.

The software architectural model is either a concrete or a reference one. The concrete software architectural model targets a specific problem domain within a specific context. It may not be used in other contexts. On the other hand, the software reference architectural (RA) model targets a specific problem domain which could be used in different contexts with or without modifications in the basic model.

The software development community (SDC) developed some quantitative methods like SAAM, ATAM, and ALMA [1] to evaluate the architectural models. These methods depend mainly on the human factor. There are other quantitative methods which provide more concrete figures about some architectural attributes like cohesion, reusability, and maintainability.

Software architects sometimes need to realize the architecture before implementation since it gives them more confidence about the pursued decisions. Accordingly, the SDC improvised some traceability and experimental methods. These methods try to link the system requirements with

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 59–77, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80505

the architectural decisions and hence may instantiate concrete architectures, generate use cases, develop prototypes, or even develop complete applications to measure the system architecture coverage against the system requirements.

A simulation prototype is an experimental prototype. ‘Simulation’ is an artificial activity that tries to imitate an operation in the real world across a period of time [2]. It could be done manually or may be automated depending on the complexity of the simulation operation scenario.

Our research work aimed to generate a software reference architecture for pervasive computing systems from a software engineering perspective (PervCompRA-SE). The reference architecture which was developed contains business and technical sides. The technical part fulfills the needs of the business part by introducing reference models for the smart object, smart environment, and the pervasive system. The baseline reference architecture in pervasive computing explains the basic structure and the behavior of the pervasive system. We used both qualitative and quantitative methods to evaluate the reference architecture. Moreover, we implemented a simulation prototype to realize the baseline architectural model for pervasive systems [3].

The paper is organized as follows: Section 2 covers the work related to the evaluation of pervasive systems. Section 3 explains our approach to build the case study. Section 4 narrates the details of the simulation story. Section 5 explains the conceptual model of the simulation prototype and section 6 gives the specification of the prototype which recognizes the story and the conceptual model. Section 7 lists the assumptions that we used to make the simulation prototype close to reality. Section 8 depicts the different simulation scenarios and Section 9 shows the prediction of the system behavior at runtime with respect to reliability and availability. Finally, section 10 concludes the paper.

2. RELATED WORK

Research efforts to evaluate software architectural models are already being applied. The existing methods are qualitative, quantitative, or experimental.

Angelov et al. [4] reported on an evaluation methodology for a reference architecture that they developed for a B2B e-contracting solution which aims to improve the contracting process between companies. The Researchers adopted the Architecture Trade-off Analysis Method (ATAM) [1] method with some variations. The authors concluded that in order to maximize the benefit from the ATAM process, then they first need to adapt the step of identifying the stakeholders properly based on the maturity level of the reference architecture, whether it is a practical or a visionary reference architecture. Second, they recommended to select a number of scenarios from different contexts, merge them, then prioritize them in a general format.

Graff et al. [5] proposed a variant from the SAAM [1] to evaluate a RA for an embedded software. Their approach is based on real-life projects in one of the leading copier manufacturers. One of the main challenges for their research work was that they needed to evaluate their RA based on concrete scenarios that can hardly attribute their design decisions to their RA which they called RACE. They decided to resolve this issue by asking a simple question while executing each scenario “What is the impact on the reference architecture?”

From another perspective, the evaluation of system architectures is seen as a straightforward task that can be achieved using quantitative figures. Madhusudanan and Prasanna [6] used metrics for pervasive systems that cover key-design aspects in pervasive systems and middleware platforms in specific. For example, the authors evaluate context-awareness for pervasive systems with respect to the number of locations, environment, user activities, time, and physical objects. They evaluate scenarios with respect to location according to the number of used locations in the selected scenario against the total number of locations in the environment. They do the same evaluation for other attributes like no. of devices and activities then build an evaluation graph.

Malik et al. [7] proposed an evaluation framework that differentiates between quantifiable and non-quantifiable characteristics of pervasive systems. Their approach considers different factors from the system, users, context, and environment. Maintainability, security and privacy, infrastructure, and integration Design factors are considered crucial evaluation factors. However, according to their evaluation, a pervasive system will not be successful if it does not meet user needs and considers user-related factors such as demographics, health, and comfort.

Bueno [8] presented a software reference architecture for component-based self-adaptive software systems. She adopted a more concrete approach to evaluate her RA by instantiating a concrete architecture and implementing a software based on it. She ran some test cases with an assumption that the quality of the application at run-time was an indicator of the quality of the architecture which was instantiated from the reference architecture.

Bogado et al. [9] introduced an evaluation framework for software architecture runtime quality attributes. The authors worked on building a discrete-event simulation model that evaluates quality attributes for a software architecture. They built a specification model using Discrete Event System Specification (DEVS) to formalize their model which was then fed into a simulator. The authors claimed that this method was useful in evaluating an architecture in the early stages of the software development lifecycle.

They described a conceptual model for evaluation. This model captures the generic behaviour of the architecture elements. It has a high level element called Architecural Element which is specialized into a Connection Mechanism and Component. The Component is further specialized into Simple Component and Composite Component. The component is the one that carries responsibilities and has a representation at runtime. The Composite Component is composed of Simple Component and Composite Component elements and its behaviour is determined by the simple components. Quality attribute values (QualityAttributeValue) are identified for responsibilities and measurements (Measure) are taken for them.

Almost every project chose a single evaluation approach to work with. We can rarely find a research project that combined different methods to evaluate a concrete architecture or an RA; although the combined view can provide useful insights for the quality of the RA. The RA represents the model with modules that could be evaluated quantifiably while its documentations could be evaluated subjectively. Moreover, very few reference models adopt the simulation prototype as a method to study the impact of the architecture on the behaviour of the system.

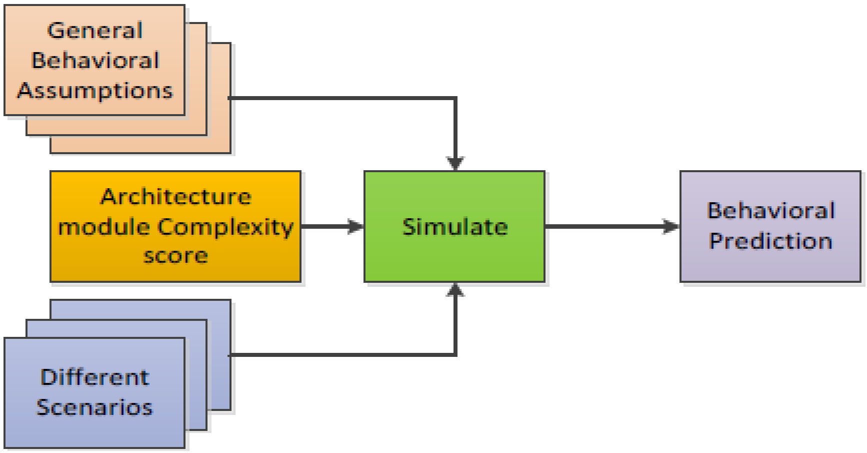

General Behavioral Assumptions
Architecture module Complexity score
Different Scenarios
Simulate
Behavioral Prediction

Figure 1. Simulation Experiment (High Level)

3. THE APPROACH

There are several methods to evaluate the structure and the behaviour of a software architecture prior to implementation. These are lead measures which reveal some issues before turning the architecture into implementation. However, some of the issues remain to be discovered after implementation. Accordingly, we need to have a lag measure in order to understand more about the system behaviour at runtime.

Hence, we assessed the reliability and availability of the architecture during runtime by running the simulation experiments. We built a conceptual model and captured state details similar to the ones mentioned in [9]. The results of the simulation were studied to propose enhancements for the baseline architecture, whenever required.

Compared to regular prototype implementations, the simulation prototype does not introduce external variables to the prototype like the hardware, programming language and network. Including these variables would have definitely impacted the final results of the experiment and is time-consuming as well. The simulation model can however clarify the requirements of the user in a virtual space which considers all constraints and quality requirements [10]. The regular prototype implementation may be more suitable for a concrete architecture. In contrast, the simulation approach is better within our scope of research, and at the same time experiments were executed in a more controlled environment. Moreover, pervasive systems are considered quite complex and they cannot follow the traditional development cycle. They need to be tested first using simulation approaches as recommended by Brink [11].

We adopted a discrete-event simulation (DES) approach which can have different states across discrete points in time. Compared to continuous event simulation, a DES simulation approach best fits operations whose states change continuously over time as per Martensson and Jonsson [12].

The real challenge for running simulation experiments to predict the behaviour of a pervasive system, or even a software system in general, is that there is insufficient historical data about similar systems in order to build a robust simulation model [9]. Usually, developers will go for assumptions and opinions from domain experts. Researchers like Roshandel et al. [13] introduced a software reliability prediction model before implementation based on the reliability of the architecture components. However, their approach requires deep knowledge about the components' design during the design phase. In our approach we ran the experiments based on the technical specifications of the sensors and also on statistics gathered and produced by earlier researchers as well as our scientific calculations for the complexity of the modules as explained in [3]. We executed different scenarios in order to provide a prediction model with a high degree of confidence (Fig. 1).

We developed the simulation prototype using the DEVS-Suite tool [14]. It was fairly easy to customize and even introduce more features to the application by introducing a simple database and building the modules using the Java language.

4. THE SIMULATION EXAMPLE

The simulation scenario that we investigated is a system that studies the quality of the sensors in a bus. There is a bus starting its trip from point A towards point B for a complete 20-hour trip. There is a location and a speed sensor installed on the bus to help the control room detect if the bus has a problem during its trip or not. The system will device its intelligence to make sense of the received data and transform them into meaningful contexts, then interpretations, then decisions, and finally actions. The bus driver will carry out different manoeuvres to stimulate the sensors. For example, he will drive normally then stop suddenly. He can drive normally then slow down suddenly. In other words, he will drive at different speeds with no alarm on when he speeds up or slows down while moving from point A to point B.

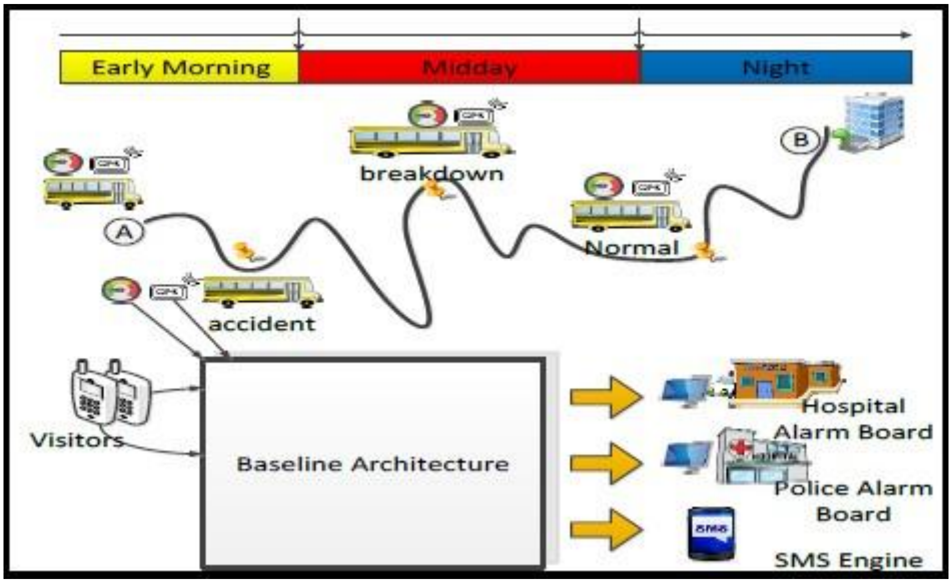


Figure 2. Bus Trip Emergency Study Simulation Example

The data generated from the sensors are classified into specific events:

- **Location Event:** is categorized based on the proximity of the bus from point A and B (At point A, Far away from point B, Midway to point B, Very close from point B, At point B).

- **Speed Event:** is categorized from an accident status point of view (Normal speed, Slowing down, Moving very slowly, Slowed down suddenly, stopped suddenly).

- **Time Event:** is categorized as (early morning, midday, and night).

The different combinations of these events generate a 3-tuple context which derives a specific interpretation. The interpretation drives a decision, which leads into some actions through the system actuators (Digital screens and SMS Gateway).

On the other hand, the system receives visitors who request services. The entities of the system may fail to achieve their duties at some time, but the autonomous error recovery of the system will work on fixing them. Moreover, the system optimization service will monitor the lifetime of the sensors to prolong their lifetime and the rest of the entities to reduce their failure rates. The whole system will be running at different modes in which there are some policies that will be applied (Fig. 2).

5. Conceptual Model

The conceptual model is an important step towards the complete implementation of the simulation prototype. It is derived from the structure of the baseline architecture model as introduced in the original research work [3]. The model is composed of Entities classified as part objects, resident objects, and visitor objects. The part objects are those modules that define the baseline architecture. The sensors and actuators are resident objects that the part modules interact with to receive data and output data. The smart objects are visitors that request services from the system on regular basis (Fig. 3).

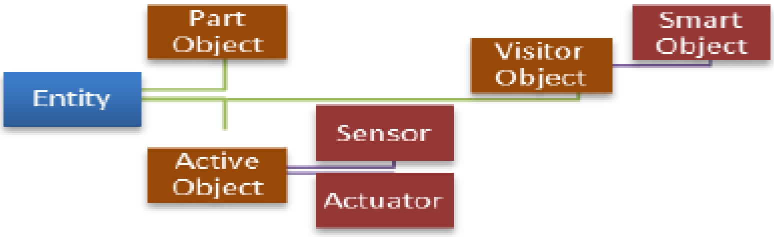


Figure 3. Simulation Conceptual Model

Entities interact with each other through their input and output ports. Every simulation module has two basic attributes (phase and tick). *Phase* represents the status of the entity and *tick* is the logical time at which the entity can accept inputs and generate outputs. All the entities are working on a tick = 10, which is equivalent to one minute, and all the entities have 4 basic phases as shown in Fig. 4:

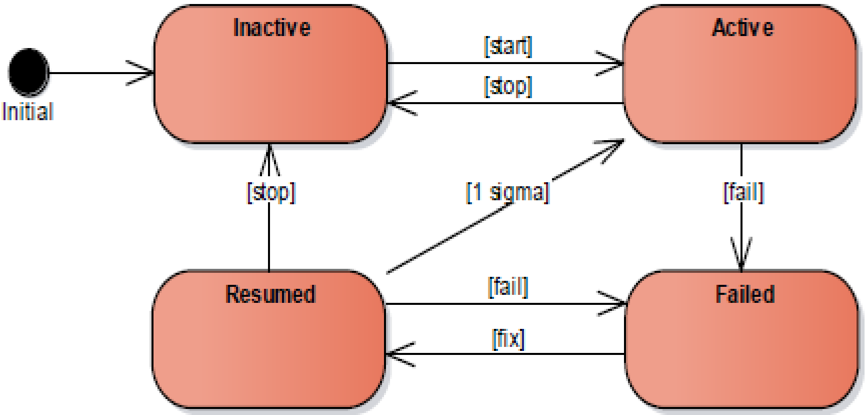


Figure 4. Simulation Module Phases

The whole simulation model can be working in one of the following modes:

1) **Runtime:** It is the normal execution scenario without changes in the settings of the entities.

2) **Assertion:** It is a normal execution scenario but with additional logging activities from these entities. They send data to the Logger to log a specific event.

3) **Security Threat:** In this mode, the whole system is threatened and needs to take some measures to protect itself. It rejects visits from new smart objects recognized as visitors and accepts visits from the trusted smart objects only. It disables the Synthesizers so that no data can be collected from the sensors.

4) **Out of Service:** During this mode the system will not be processing sensor signals, will not accept visits, and will not fetch user profiles from the Repository Manager. The system will still keep recording sensor data and when the system returns to one of the other three modes, then it can fetch the data and work on it.

The state of an entity at any point of time is defined using the 6-tuple (P, AI, AO, L, F, M):

1) **Phase (P):** it is the phase of the entity where P is one of the phases in the set {Active, Inactive, Failed, Resumed}.

2) **Accumulated Inputs (AI):** it is the number of received input requests for all the input ports
$\boldsymbol{AI = \sum_{i=0}^{n} count\ (input_i)}$

3) **Accumulated Outputs (AO):** it is the number of submitted outputs for all the output ports
$\boldsymbol{AO = \sum_{i=0}^{n} count\ (output_i)}$.

4) **Lifetime (L):** is the lifetime indicator of the entity which takes a value from 0-100. 100 indicates that it is healthy and fully powered, and 0 indicates that it is dead. It is an optional state attribute for part objects

5) **Failures (F):** it is the counter of the failures. It is ceiled by a maximum threshold. The counter will reset to 0 after reaching the threshold. It is an optional state attribute for active objects.

6) **Mode (M):** It is the mode of the system where M is one of the modes in the set {*Runtime, Assertion, Out of Service, Security Threat*}.

6. MODEL SPECIFICATION

The main building modules of the simulation project are derived from the baseline architecture as specified in [3]. In addition, we added more modules to make the prototype more controllable and fulfilling for the simulation example as well.

The prototype starts using the *Simulation Starter* module which is responsible for starting, stopping, changing execution modes, and dumping statistics about the simulation runs. The *Policy Manager* is responsible for applying the system policy on the modules according to the mode of execution. The sensors of the prototype are the *Speed Sensor* and the *Location Sensor*. Both send their data to *Sensor Synthesizers* which receive the data and generate the sensor signals after checking their correctness based on a standard deviation of expected error factors.

The prototype uses two digital boards (*Hospital Alarm Board* and *Police Alarm Board*) and an *SMS Engine*. They are used to send alarms and notifications about accidents that may occur during the simulation. They are the actuators that fulfil actions in order to adapt to the changes in the context of the experiment.

The *Repository Manager* is responsible for storing the synthesized sensors' data in a 3-tuple format including the time, location and the speed. It stores also the profile of the users, which is managed by the *Profile Manager*, and the visits made by the smart objects.

The system makes sense of the 3-tuple data format by transforming it into a 3-tuple context through the *Event Handler*. The system then employs the Interpretation Manager to interpret the 3-tuple context. The *Decision Manager* uses the interpretation to make a decision which triggers actions through the actuators.

The systems simulates the visits of the users through the *Smart Object* which sends visits to the system similar to the attendance models of the employees as it is expected to have more users join the system during the day and the trend decreases slowly where the disjoin trend increases by the end of the day. The *Device Manager* is responsible for registering the joins and leaves of the smart objects. The *Service Manager* receives requests from the visitors and fulfils them through the different modules of the system like the *Repository Manager*, actuators, and the sensors.

Every service has an authorization level based on the smart object type, *visitor* or trusted smart object. If the smart object requests a service that has an authorization level not suitable for its type, then the *Service Manager* rejects the request. The *Risk Handler* is responsible for studying the requests from the smart objects to join the system and puts it on the proper status (*visiting, trusted, prohibited*, or *rejected*). It is also responsible for handling the certificate requests sent from the joining smart objects.

The *Fault Manager* is responsible for handling faults that cause part objects to be out of service. For the sake of consistency and better tracking in the simulation model, the *Fault Handler* is responsible also for failing the modules. It is important to note that the probability of part object failure increases based on its complexity as shown in equation (1) [3]:

$$weight = Round(\frac{r * d}{\sum_{i=0}^{15} r_i * d_i} * 100) \qquad (1)$$

Where *r* is the number of satisfied requirements by the part object and d is the number of input and output dependency relationships for the part object. The equation derives the faults from the satisfied requirements, which could be translated as internal part object capabilities and the dependency relationships with other part objects. It is then divided by all the weights of the modules and multiplied by 100 to get a percentage. The weight is rounded off after that.

The *Optimization Manager* is responsible for monitoring the failure rates of the modules and the lifetime status as health performance indicators for the sensors, actuators, and the part objects and takes decisions to recover their performance. The *Resource Manager* receives a request from the *Optimization Manager* to allocate a resource for a nominated part object, or sensor. If the request is to reduce failures, then the *Resource Manager* will select a resource, which could be a hardware or a software, randomly from a set of resources reserved for the part objects only, if not already allocated. The part object receives the resource which gives it a limited protection from failure through a pre-defined period of time, e.g. 100 ticks. Accordingly, if the *Fault Handler* decides to fail a part object that has a resource allocated for it, the part object will ignore this fail message.

If the request is to recover the lifetime of the sensor, then the *Resource Manager* will select a resource randomly from a set of resources reserved for the sensors only, if not already allocated. The battery resources increase the lifetime of the hardware instantly by a specified lifetime value [15].

The data collected by the *Repository Manager* and the *Logger* is analysed by the *Analysis Manager* which shares some of its knowledge with the external entities represented as a module called *Interested Community*.

7. SIMULATION ASSUMPTIONS

There are some important assumptions that had to be designed within the probability model of the simulation prototype. They are derived from real data concerning software and hardware components. These assumptions are embedded in the simulation prototype as settings which could be modified as needed to produce numerous simulation scenarios. In order to make the simulation scenario very close to reality, we made some assumptions derived from actual statistics.

The sensors are assumed to be running on batteries that deplete gradually according to the rate of generated sensor data. They start with an initial capacity of 100%, and decrease gradually by X% (e.g. 0.5%) with every activity. On the other hand, part objects and actuators are running on permanent power, but they may experience random failures every now and then.

We investigated the technical settings of some sensors (battery lifetime, and accuracy for the time pulse signal). The ideal settings for the sensors are as follows:

1) **Speed Sensor**: It is assumed to have a 15 hours battery lifetime in normal conditions [16]. We assume the minimum hours for the battery is 7h12m and maximum is 18h42m [17] and the probability of failure between 10-7 and <= 10-8 per hour [18]. Our settings are derived from two speed sensor products.

2) **Location Sensor:** The same battery lifetime of the speed sensor is assumed here as well for the location sensor [17]. The horizontal position accuracy has a standard deviation of 0.35 meters [19]. The assumptions are derived from one product.

We assume that the probability of failure for the SMS Engine is 0.05 as per the research on the reliability of short messaging [20]. The failure rate of the SMS Engine as an active object is independent from the part objects of the system.

We assume that the failure rate for the digital screens installed in the Police department and the Hospital is 0.03 due to product defects as stated by Shaw [21]. Given other external factors like scheduled maintenance, power supply cut-off, and software failure, then we can safely increase the failure probability to 0.05. The digital screen failure rate is quite independent from the failures of the part objects.

Part objects are assumed to be running on different servers from the same manufacturer and the same manufacturing year. We assume that the best probability of total failure for the part object at any minute is 0.05 based on estimates from [22] [23]. On the other hand, another interesting research, by YAN [24], shows that the reliability of a pervasive system can be less than 0.5. So, we assign 0.5 as the worst probability. The average in this case will be 0.275. The values for the control variable (Part Object Failure Optimization Threshold) as shown in Table 1 are based on experience with IT support units in the Telecom industry.

Pervasive systems, or IoT systems, are highly vulnerable to security threats [25]. Moreover, systems usually go out of service due to planned maintenance or unplanned outages. It is noticed that the cost of maintaining a system is in continuous increase since the end of the last century. This is basically due to the increased number of developed software applications and their increased complexity [26]. Moreover, administrators dump logs from the system for monitoring purposes all the time. Accordingly, we assume that the system will be running in normal mode most of the time (64-70%) and that there is 30-36% probability that it will be running in one of the other abnormal modes (Assertion, Out of Service, or Security Threat).

The *Optimization Manager* checks the status of the battery if it reaches 40% of its capacity [27] on average. We assume that the mean time of repair for the part object is shorter than the mean time between failures [28]. We assume that the more complex is the part object, the higher the probability of failure, and the less complex is the part object the faster we can get a repair [29] [30].

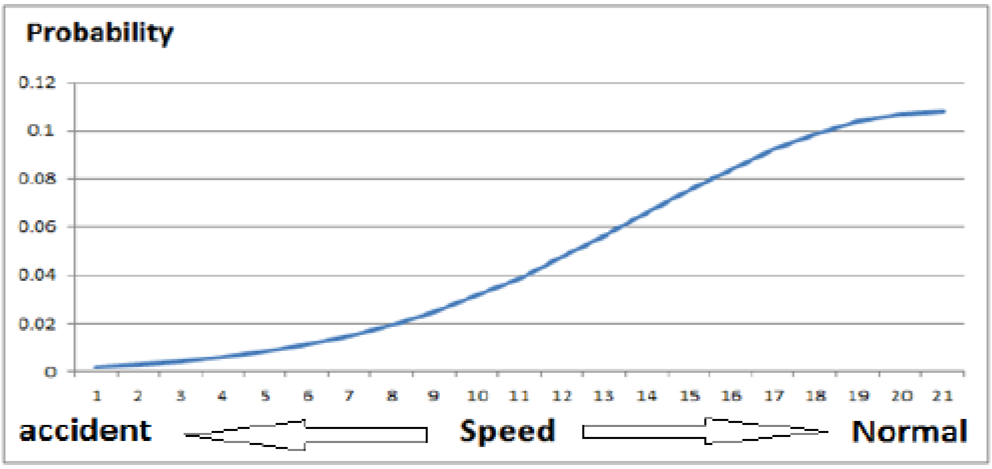


Figure 5. Speed normal probability function

We also assume that the probability of having a normal trip is normally distributed around normal driving and that the accidents are rare (bell curve shape) with very low probability as reported by some studies around accidents in the USA [31] [32]. We executed different runs to generate values by the Gaussian function, and optimized the standard deviation σ in order to get a bell curve that fits the probability distribution model for normal driving and accidents. We used average = 21 and σ = 7. Anything greater than 3σ and less than -3σ is set to 0, otherwise values between average and 3σ are mirrored to be in the range of (0, average) as shown in equation 2.

$$RN = Gaussian\ Random\ Number * \sigma + average$$
$$RN = \begin{cases} \mathbf{0}, & RN > 3\sigma\ or\ RN < -3\sigma \\ \boldsymbol{RN - 2*(RN - average)}, & RN > average\quad and\ R < 3\sigma \\ \boldsymbol{RN} & \end{cases} \qquad (2)$$

The function generates a semi-bell probability shape as shown in Fig. 5 where 0 is an indication for an accident and 21 is an indication for a normal driving speed.

We assume that the visits of the smart objects are tightly coupled with the visitors' behaviour to the attendance of employees in a workplace. It is assumed that the human visitors' trend has peak visits during the early morning and decreases through the day. Accordingly, the smart objet disjoins the system across the day but there is a peak at the end of the day when visitors start to leave the school. During this time, visitors can request services in a normal distribution where the most aggressive period is at midday (Fig. 6).

The simulation model generates new join requests from smart objects as shown in equation/algorithm (3) [3]. We assume an average of 6 visits at a time and σ = 3. It generates a large number of visits, not exceeding 7000, if the simulation model will execute for 1500 ticks. The same algorithm will generate disjoin requests but at very small rates in the early day time and increasing by the end of the day.

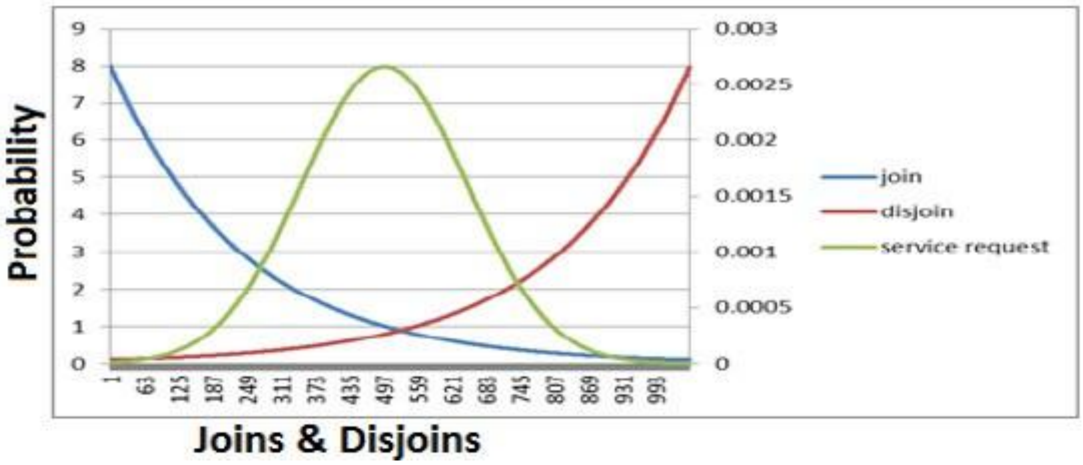


Figure 6. Smart Objects join/disjoin behavior during the simulation

```
Loop from $tick = 1$ to $N$
    $V = Gaussian\ Random\ Number * \sigma + average$
    $V = \begin{cases} \mathbf{0}, & V > 3\sigma \text{ or } V < -3\sigma \\ \mathbf{V - 2 * (V - average)}, & V > average \text{ and } V < 3\sigma \\ \mathbf{V} \end{cases}$    (3)
    $join\ requests = round(V * \left(1 - \frac{tick}{N}\right))$
    $disjoin\ requests = round(V * \left(\frac{tick}{N}\right))$
End Loop
```

The assumptions that we presented are accurate to our best knowledge and based on credible references. We evaluated the best and worst values in order to use them in our simulation experiments. They are all derived from the same sources as summarized in Table 1, given that the average may not represent a calculation from the best and worst values.

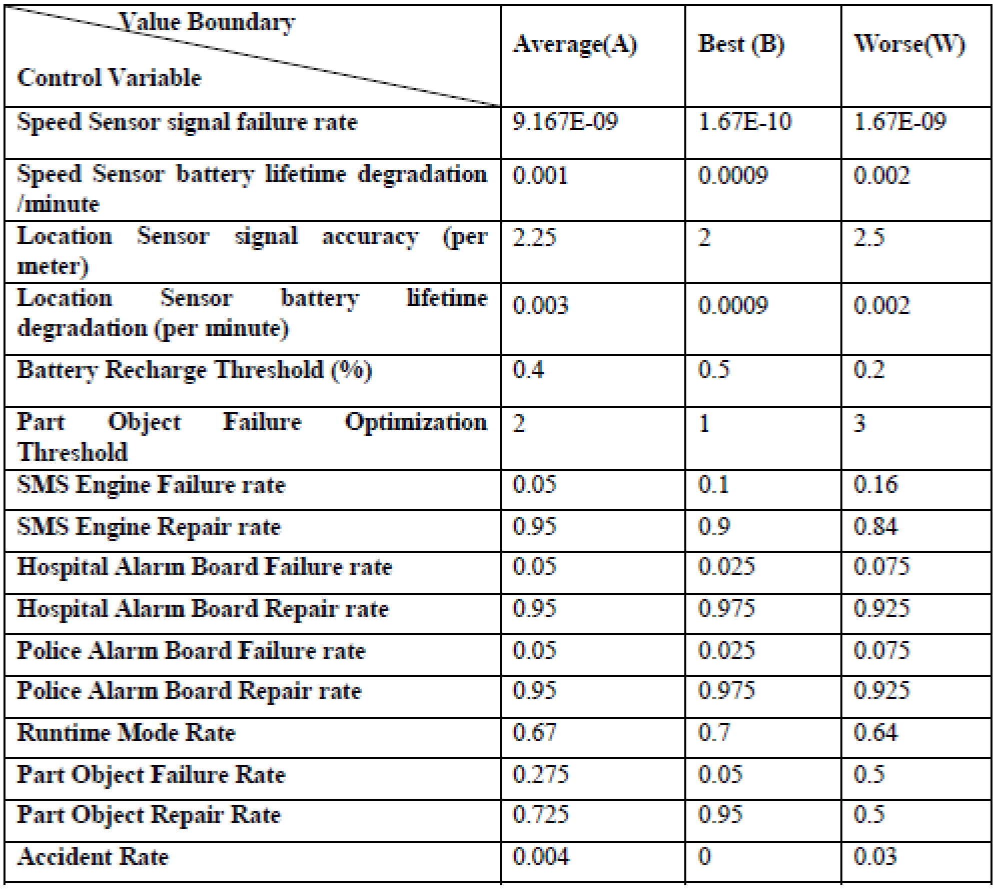
8. EXPERIMENTATION SCENARIOS

In order to provide an acceptable prediction model for the reliability and availability of the PervCompRA-SE, we implemented different simulation scenarios. We calculated the Mean Time between Failures (MTBF) and Mean Time to Repair (MTTR) in order to calculate the reliability and the availability scores of the simulation scenario.

There is the perfect scenario which assumes that the bus starts from point A to point B and that the system completes processing all the sensor data on time *Tp*. There are no failures in the system and the smart objects' requests are all satisfied, and batteries do not deplete. The perfect scenario will be used for benchmarking purpose. This scenario does not require a *Fault Handler* nor an *Optimization Manager*. It will be executing always in the *Runtime mode*.

There is the **Normal Hybrid Modes** scenario which assumes the values in Table 1. There will be faults and repairs in that scenario. The scenario introduces more disturbances to the normal flow of the execution cycle by introducing changes in its execution modes. It is expected that it will take a longer time than scenario 2. There will be 3 runs for each category of values (Best, Average, and Worst). It should finish at time ($Tp + \Delta Tnb + \Delta Tnbh$), ($Tp + \Delta Tna + \Delta Tnah$), and ($Tp + \Delta Tnw + \Delta Tnwh$), respectively. We will assume a fixed number of resources (Rn = 12) across all the runs.

Table 1. Assumed values of Control Variables



Value Boundary Control Variable	Average(A)	Best (B)	Worse(W)
Speed Sensor signal failure rate	9.167E-09	1.67E-10	1.67E-09
Speed Sensor battery lifetime degradation /minute	0.001	0.0009	0.002
Location Sensor signal accuracy (per meter)	2.25	2	2.5
Location Sensor battery lifetime degradation (per minute)	0.003	0.0009	0.002
Battery Recharge Threshold (%)	0.4	0.5	0.2
Part Object Failure Optimization Threshold	2	1	3
SMS Engine Failure rate	0.05	0.1	0.16
SMS Engine Repair rate	0.95	0.9	0.84
Hospital Alarm Board Failure rate	0.05	0.025	0.075
Hospital Alarm Board Repair rate	0.95	0.975	0.925
Police Alarm Board Failure rate	0.05	0.025	0.075
Police Alarm Board Repair rate	0.95	0.975	0.925
Runtime Mode Rate	0.67	0.7	0.64
Part Object Failure Rate	0.275	0.05	0.5
Part Object Repair Rate	0.725	0.95	0.5
Accident Rate	0.004	0	0.03

There is the **Normal No-Optimization** scenario which aims to predict the behavior of the technical model without the optimization mechanisms (*Optimization Manager*, *Resource Manager*). There will be 3 runs for each category of values (Best, Average, and Worse). It is expected that the processing time for this scenario will take additional time ($\Delta Tnoop$) for every group of runs than scenario 3. It is expected also that the MTBF will increase more than what is recorded in scenario 3.

There is the **Normal resource-optimized** scenario which aims to predict the impact of the number of resources on the system reliability. The scenario will show the impact of the *Optimization Manager* and the *Resource Manager* on the number of faults that the system may encounter. It is also expected to see some decrease, ($-\Delta T_r = 4$), ($-\Delta T_r = 8$), and ($-\Delta T_r = 12$), in the processing time relevant to the number of resources, than scenario 3 and increased time between failures ($MTBF + \Delta T_f$). There will be 3 runs for each category of resources using the Average control variables. It is important to note that the last scenario variation is the same as scenario 3 with Average control variables.

Finally, there is the **Extreme** scenario which aims to predict the behavior of the technical model under extreme conditions. We will use the extreme values, which exceed the boundaries of the best and worst, to run two categories of runs (Extreme Best and Extreme Worst). A fixed number of resources as in scenario 3 (R_{ex} = 12) is assumed here. The best and worst values in Table 1 are considered as the standard lower and upper boundaries. We built a simple capability model to stretch the best and worst boundaries as follows [33]:

1. We calculate the average from the lower and upper bounds.
2. Calculate the standard deviation (σ).
3. We calculate the minimum value, whether it is best or worst, as (min=average-3* σ).
4. We calculate the maximum value, whether it is best or worst, as (max=average+3* σ).
5. If the value exceeds the logical or physical limits, then it is set to the maximum possible value.

9. ANALYSING THE RESULTS

A reliable system is a system that can perform its assigned functionality with a high probability during a specified period of time and within specific design constraints [13]. A reliability measurement is a function in MTBF and gives a score between 0 and 1 [28] as shown in equation (4) [3]. On the other hand, software availability is the probability of the uptime of the system. It is a function of MTBF and MTTR [28] as shown in equation (5) [3]. For example, if we measure the availability of a website during a year and it is 0.99, then it means that the system downtime was (3.65 days) calculated as ((1-availability) x 365). MTBF measures the average time between successive failures without considering the time taken to repair the system in order to reflect its ability to fulfil its duties. If a system's reliability is 0.99, it means that the system is expected to run successfully from time 0 to time *t* with probability 99%.

$$Reliability = \frac{MTBF}{MTBF + 1} \qquad (4)$$

$$Availability = \frac{MTBF}{MTBF + MTTR} \qquad (5)$$

The experiments show some facts about the technical baseline architecture:

1) The experiments predict the reliability of the architecture in the worst case as 96.86% and the availability as 90.89%.

2) In the extreme worst cases both reliability and availability measurements decrease noticeably as reliability becomes 92.62% and availability deteriorates to 31.83%.

3) On average the system availability is 95.77% and reliability is 98.08% if we exclude the Perfect and extreme cases.

4) In the best cases, the system availability is 99.79% and reliability is 99.9%.

The results show that that there are variations in processing time among all the scenarios. We predict an average of 2% additional processing time as overhead from the last sensor input calculated against the perfect scenario (Fig. 7). The results show that the resource optimization technique that we adopted is working reasonably. The experiments show an average of 3.09% immunity from failures across all the scenarios. As the resources allocated increase, the immunity of failure provided to the system increases as well. In general, the scenarios show that the processing time increases as the working conditions get worse.

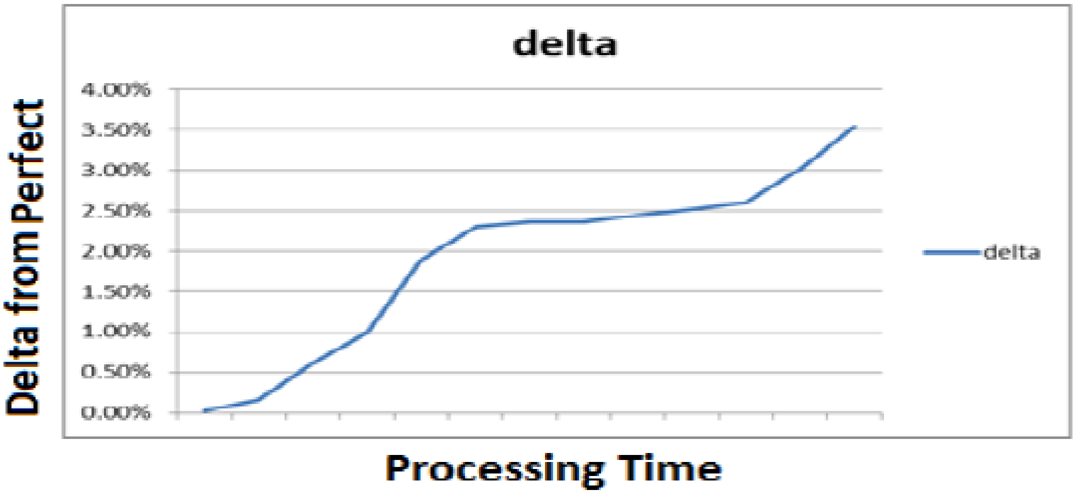


Figure 7. The processing time overhead for the simulation scenarios compared to the perfect scenario

10. CONCLUSION

The simulation case study presented in this paper was used to evaluate a baseline architecture as part of an overall evaluation of a reference architecture. There was a hypothetical example (bus story) to track the behaviour of the architecture. We assigned roles and responsibilities for the modules of the baseline architecture through a specification exercise in addition to adding other modules to introduce more controls to the simulation prototype. The prototype adopted some probability models through assumptions inferred from collected statistics about software and hardware components. The simulation prototype is then executed in different scenarios. The results were then analysed and the reliability and availability of a system adopting this baseline architecture are shown to be over 90%.

There are modules that we did not expose to failures and repairs during our simulation exercise (Repository Manager, Logger, Fault Handler, and Synthesizers). These are common modules for all the other modules in the system. They do not need a simulation exercise to understand that a single failure in the Repository Manager will hinder the overall stability of the whole system. It is very clear also that the Logger can impact the overall performance if it is not responsive. Moreover, the Fault Handler is designed to respond to failures, and it is essential to make it more reliable and available than other modules. Finally, the Synthesizer is either a part of the sensors and actuators hardware or it is at a low-level software layer that the sensors and actuators must interact with. If this layer fails, then the data may be corrupted.

Although the analysis shows positive results about the reliability and the availability of the architectural model, the prediction is an initial estimation which will definitely change in a real environment. This should be a continuous improvement process by fetching real numbers about the systems' performance during runtime in order to give more accurate predictions about the probability of the system failure.

It is quite beneficial for the architects to have an accessible simulation package for the reference architecture model containing configurable settings for all the control variables. The simulation package should help the architect represent the different use cases as he/she builds a real-life scenario using the PervCompRA-SE.

ACKNOWLEDGEMENTS

A special gratitude goes to Professor Veronica Bogado in the *Departamento Ingeniería en Sistemas de Información*, UTN FRVM for providing the necessary material that helped with this simulation project.

REFERENCES

[1] M. A. Babar and I. Gorton, "Comparison of scenario-based software architecture evaluation methods," in 11th Asia-Pacific Software Engineering Conference, 2004, pp. 600–607.

[2] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, Discrete-Event System Simulation, 5th ed. Prentice Hall, 2010.

[3] O. M. Khaled, "Pervasive Computing Reference Architecture from a Software Engineering Perspective," Ph.D. Dissertation, The American University in Cairo, Cairo, Egypt, 2017.

[4] S. Angelov, J. J. M. Trienekens, and P. Grefen, "Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case," in Software Architecture: Second European Conference, ECSA 2008 Paphos, Cyprus, September 29-October 1, 2008 Proceedings, R. Morrison, D. Balasubramaniam, and K. Falkner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 225–240.

[5] B. Graaf, H. van Dijk, and A. van Deursen, "Evaluating an Embedded Software Reference Architecture — Industrial Experience Report —," in Ninth European Conference on Software Maintenance and Reengineering, 2005, pp. 354–363.

[6] J. Madhusudanan and V. Prasanna Venkatesan, "Metrics for Evaluating Pervasive Middleware," Int. J. Intell. Syst. Appl. IJISA, vol. 6, no. 1, pp. 58–63, Dec. 2013.

[7] Y. Malik, M. Soliman, and B. Abdualrazak, "Towards an Evaluation Framework for Pervasive Computing System," in International Conference on Modeling, Simulation and Visualization Methods (MSV), Las Vegas, USA, 2011, p. 8.

[8] L. C. Bueno, "A Reference Architecture for Component-Based Self-Adaptive Software Systems," Department of Information and Communication Technologies Faculty of Engineering, ICESI University, Cali, Columbia, 2012.

[9] V. Bogado, S. Gonnet, and H. Leone, “A Discrete Event Simulation Model for the Analysis of Software Quality Attributes,” CLEI Electron. J., vol. 14, no. 3, Dec. 2011.

[10] S. P. Miller, “Proving the Shalls: Requirements, Proofs, and Model-Based Development,” in 14th IEEE International Requirements Engineering Conference (RE’06), 2006, pp. 266–266.

[11] S. S. Brink, “Enabling Architecture Validation in the Analysis Phase of Developing Enterprise or Complex Systems using Enterprise Architecture Simulation Environment (EASE),” in MILCOM 2007 - IEEE Military Communications Conference, 2007, pp. 1–8.

[12] F. Mårtensson and P. Jönsson, “Software Architecture Simulation – a Continuous Simulation Approach,” Master’s thesis, Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Sweden, 2002.

[13] R. Roshandel, N. Medvidovic, and L. Golubchik, “A Bayesian Model for Predicting Reliability of Software Systems at the Architectural Level,” in Proceedings of the Quality of Software Architectures 3rd International Conference on Software Architectures, Components, and Applications, Berlin, Heidelberg, 2007, pp. 108–126.

[14] “DEVS-Suite,” Arizona Center for Integrative Modeling & Simulation, 2.0.

[15] B. Lawson, “A Software Configurable Battery,” presented at the EVS26 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium (EVS26), Los Angeles, California, 2012.

[16] “Speed Sensor Edge 520.” [Online]. Available: https://www8.garmin.com/manuals/webhelp/edge520/EN-US/GUID-F50056D5-6DC6-43D2-81A6-61095620E142.html. [Accessed: 21-Apr-2017].

[17] K. Byrne, “Best phone battery life 2016: Top smartphones tested,” Expertreviews, 02-Nov-2016.

[18] “SIL3 Speed Sensors.” [Online]. Available: http://www.jaquet.com/site/assets/files/1218/flyer_sil3_a4_en.pdf. [Accessed: 21-Apr-2017].

[19] “NEO-6 u-blox 6 GPS Modules.” [Online]. Available: https://www.u-blox.com/sites/default/files/NEO-M8_DataSheet_(UBX-13003366).pdf.

[20] X. Meng, P. Zerfos, V. Samanta, S. H. Y. Wong, and S. Lu, “Analysis of the Reliability of a Nationwide Short Message Service,” in IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, 2007, pp. 1811–1819.

[21] M. Shaw, “Reducing LCD TV Warranty Claims Through an Organized and Aggressive Approach to Sub-Assembly and Full LCD TV Assembly Accelerated Stress Testing,” presented at the International Applied Reliability Symposium, EUROPE, Milan, Italy, 2010.

[22] “Frequency of server failure based on the age of the server (per year).” [Online]. Available: https://www.statista.com/statistics/430769/annual-failure-rates-of-servers/. [Accessed: 21-Apr-2017].

[23] O. Bäckström, J.-E. Holmberg, M. Jockenhövel-Barttfeld, M. Porthin, A. Taurines, and T. Tyrväinen, “Software reliability analysis for PSA: failure mode and data analysis,” Nordic Nuclear Safety Research (NKS), NKS-341, ISBN 978-87-7893-423-9, Jul. 2015.

[24] L. YAN, “Applying Model Checking to Pervasive Computing Systems,” Ph.D. Dissertation, Department of Computer Science, School of Computing. National University of Singapore, Singapore, 2014.

[25] D. Storm, “Of 10 IoT-connected home security systems tested, 100% are full of security FAIL,” COMPUTERWORLD, 11-Feb-2015.

[26] J. De Vries, C. Burki, and B. De Vries, “How to save on software maintenance costs,” Omnext, Nov. 2014.

[27] S. Jary, “How to properly charge a phone’s battery: stop charging from zero to 100% and other tips,” TechAdvisor, 26-Jul-2016.

[28] H. Pham, System Software Reliability. Springer London, 2006.

[29] Y. Liu and I. Traore, “Complexity Measures for Secure Service-Oriented Software Architectures,” in Predictor Models in Software Engineering, 2007. PROMISE’07: ICSE Workshops 2007. International Workshop on, 2007, pp. 11–11.

[30] J. Iqbal, D. S.M.K.Quadri, and T. Rasool, “On Way to Acquiring Reliability Growth in Software Systems,” Int. J. Comput. Appl., vol. 24, no. 7, pp. 33–36, Jun. 2011.

[31] “Traffic Safety facts Research Note,” U.S. Department of Transportation, National Highway Traffic Safety Administration, Aug. 2016.

[32] D. Toups, “How many times will you crash your car?,” Forbes, 27-Jul-2011.

[33] D. S. Moore, G. P. McCabe, and B. A. Craig, Introduction to the practice of statistics : extended version, 6th ed. New York: W.H. Freeman, 2009.

AUTHORS

Osama M. Khaled received his Bachelor, Masters and PhD degrees in Computer Science from the American University in Cairo in 1998, 2004, and 2017 respectively. Osama works in the software development and telecommunication industry since 1998 and acts as a lecturer and consultant in software engineering as well. His active research areas are in software engineering, software architecture, software modelling, business analysis, and software programming. Osama is the author/co-author of 12 publications in international journals and conference proceedings.

Hoda M. Hosny is a Professor of Software Engineering and has been teaching at the American University in Cairo since 1985. She started her teaching career in the University of California, Davis (UCD) in 1981. She received her Masters degree in Computer Science from UCD in 1984 and her Ph.D. from Leeds University, UK, in 1991. She served as an IT Specialist, Consultant and Trainer at a number of National and International Institutions and was invited to lecture at 4 other universities in Cairo. She is the author/co-author of more than 50 publications in international journals and conference proceedings.

Mohamed Shalan is an Associate Professor (with tenure) at the Department of Computer Science and Engineering, the American University in Cairo. He received his Ph.D. in computer engineering from the Georgia Institute of Technology (GaTech) in 2003. He received his B.Sc. and M.Sc. in computer and systems engineering from Ain Shams University, Cairo, Egypt in 1993 and 1997. His research interests are in the area of computer engineering, with focus on embedded systems, digital design, energy-efficient computing systems, and electronic design automation. Professor Shalan has over 30 refereed conference and journal papers. Also, he holds 2 US patents.

INTENTIONAL BLANK

THE PREDICTION OF STUDENT FAILURE USING CLASSIFICATION METHODS: A CASESTUDY

Mashael Al luhaybi, Allan Tucker and Leila Yousefi

Computer Science Department, Brunel University, London, UK

ABSTRACT

In the globalised education sector, predicting student performance has become a central issue for data mining and machine learning researchers where numerous aspects influence the predictive models. This paper attempts to apply classification algorithms to evaluate student's performance in the higher education sector and identify the key features affecting the prediction process based on a combination of three major attributes categories. These are: admission information, module-related data and 1st year final grades. For this purpose, J48 (C4.5) decision tree and Naïve Bayes classification algorithms are applied on computer science level 2studentdatasets at Brunel University London for the academic year 2015/16. The outcome of the predictive model identifies the low, medium and high risk of failure of students. This prediction will help instructors to assist high-risk students by making appropriate interventions.

KEYWORDS

Prediction, classification, decision tree, Naïve Bayes, student performance

1. INTRODUCTION

In recent years, there has been an increasing interest in applying data mining algorithms in various fields such as medicine, marketing, education, engineering so forth, due to its benefits in transforming huge amount of such data into useful knowledge. Data mining (DM), or in other words Knowledge Discovery in Databases (KDD), can be defining as a multi-disciplinary field in which several computing paradigms converge: decision-trees, artificial neural networks, rule induction, instance-based learning, Bayesian learning, logic programming, statistical algorithms, etc. The most well-known data mining techniques are Clustering, Classification, Association rule mining and Description and visualisation [1].

The growing availability of data in educational databases attracts many researchers to analyse and evaluate such data to enhance education and provide optimal solutions for associated issues. This emerging discipline is called Educational data mining (EDM) where we apply data mining (DM) techniques or develop new DM methods to explore educational data in order to understand student's learning process and their outcomes [2].

Within the education field DM seeks to analyse students learning by developing approaches that merge student's data and data mining algorithms to benefit the students and enhance their

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 79–90, 2018. © CS & IT-CSCP 2018   DOI : 10.5121/csit.2018.80506

learning process. However, student's performance plays a crucial role in students' academic achievement. The final grades obtained by the students throughout his/her academic study inspire their future. Therefore, it becomes essential to determine whether the students will pass or fail the module. If the predictive model can characterize the students with high risk of failure prior the examination then the academics can provide extra effort to improve students' performance and assist them to pass the module or obtain higher results.

In this connection, this study seeks to address the following:

- Factors affecting the prediction of the high risk of failure of students in higher education institutions and universities,

- Predictive data mining models using classification algorithms based on level 1 student final grades, modules related data and students admission datasets.

2. RELATED WORK

Researchers have been increasingly attempting to analyse students' datasets using data mining and machine learning algorithms in order to understand how students learn and to ultimately increase the performance of students and the quality of learning. However, a considerable amount of literature has been published on predicting the performance of the students based on different factors and attributes. These are summarized in Table 1.

Cortez and Silva [3] conducted a study to predict the performance of secondary school students based on demographic, social and past school grades. By means of Classification and Regression algorithms (Decision Trees, Random Forest, Neural Networks and Support Vector Machines), it was found that the past evaluation of the students were highly influenced with their performance. Also, there were other factors that correlated with the students' academic performance (such as: number of absences, parent's job and education, alcohol consumption).

Preliminary work on mining student datasets to predict their performance was undertaken by Al-Radaideh et al. [4]. They applied Classification algorithms ID3, C4.5, and Naïve Bayes on student's data that obtained via questionnaire for the academic members of C++ programming course at Yarmouk University, Jordan. The attributes included in this study were students demographic and tutors related data such as degree, gender and affiliated department. Weka mining tool was used in this investigation to develop the predictive models. The outcome expressed the correlation between the high school grades and students' academic performance.

Aher and L.M.R.J. [5] attempted to analyse the examination performance of final year students for undergraduate module using Weka mining tool. The algorithms Association Rule, Classification (ZeroR), Prediction and Clustering (DBSCAN) were applied on student's examination data to study the possibility of applying data mining on educational systems. The outcome of their result indicates the usefulness of data mining algorithms for higher education data especially to improve the students' performance.

A comparative analysis has been conducted by Yadav and Pal [6] to predict the final exam performance for engineering students . They applied ID3, C4.5 and CART decision trees algorithms on student's datasets that include personal, social, psychological and environmental factors for the prediction task. The obtained results reveal that C4.5 decision tree prediction model gives better result than ID3 and CART with accuracy of 67.77% for identifying the weaker students before the examination and that help them to improve their study for better exams results.

Another study was conducted by López et al. [7] to predict the final grades of the students based on their participation in the online forum using Weka mining tool. By means of Clusteing algorithms (EM, FarthestFirst, HierarchicalClusterer, sIB, SimpleKMeans, and XMeans) they found that students participation in the course forum is a predictive factor for predicting student final grade in a module.

As shown in Table 1, researchers have attempted to analyse students demographic, social and assessment data to predict the slow learning students in order to improve their performance and reduce failure rate prior the exam [8][11]. Also, there are several studies which compared Naive Bayes method with other classification methods to classify the students and identify their abilities, interests and weaknesses [9][10].

Table 1. Accuracy results based on Decision Tree and Naïve Bayes methods

Method	**Attributes**	**Accuracy**	**Authors**
Decision Tree	Past school grades (first and second periods), demographic and social data	76.70%	[3]
	High school dataset (Demographic, Personal Data and Admission data)	69.73%	[11]
	personal, social and psychological data	67.77%	[6]
	Personal and pre-university data	65.94%	[9]
	Demographic, personal and psychological data	61.53%	[10]
	Demographic, personal and tutors related data	38.05 %	[4]
Naïve Bayes	Demographic, CGPA and course assessments data	73%	[8]
	Demographic, social data and past grades (first and second periods)	65.13%	[11]
	Demographic, psychological and environmental data	63.59%	[10]
	Demographic and pre-university data	58.10%	[9]

Bayesian classification method was applied by Bekele and Menzel [12] to predict students' performance based on values of social and personal attributes. The empirical result revealed that Bayesian network classifier is a valuable method for predicting the students having satisfactory, or above/bellow satisfactory performance.

Another Bayesian classification method (in particular Naïve Bayes) was modelled by Bhardwaj and Pal [13] to predict the slow and the high learner's students. The study conducted on 300 student records for BCA module (Bachelor of Computer Applications) from five colleges at Awadh University, Faizabad, India. The attributes included in this investigation were demographic, academic and socio-economic that obtained from students questionnaire and the database of the university. By means of Naïve classification approach, it was stated that student's performance in university level is dependent on Senior Secondary Examination grades, students living location, teaching mode and other potential factors such as (Mother's Qualification, Students Habit, Family annual income and family status).

However, the predictive data mining model presented in this paper is different from what excites in the literature as it does not involve social, psychological, environmental and personal factors to predict the academic performance of the students, it based on a combination of three data categories which are admission, module-related data and student's level 1 final grade.

3. DATA MINING PROCESS

In the educational sector, student overall grades of the Modules is an important factor to determine whether the student pass or fail the Module. The overall grade is calculated by adding the student assessment grades, course activities and final examination results. Therefore, we performed steps to predict students at high risk of failing the Module based on their final or overall grades and other aspects. These steps are as follow:

3.1. Data selection and Pre-processing

This study considers students and modules data obtained from the Admission and the Department of Computer Science databases at Brunel University London, UK. The integrated data considered in this investigation could be categorised into three categories, are as follows:

I. **Admission Data** the data relating to students information when they register at the university such as Student Enrolment Status, Student Route name, Fee Status, Student Mode of studying, Qualification on Entry, Location of Study, previous institution … etc (see Table 2)

II. **Level 1 Final Grades** the overall grades for all level 1 modules that were taken by Computer Science Students in the first year which are:
Information Systems and Organisations
Logic and Computation
Level 1 Group Project Reflection
Data and Information Assessment
Software Design
Software Implementation Event
Fundamental Programming Assessment

III. **Module-Related Data:** the data for the predicted module such as Module teaching mode, Tutor Code, Tutor Name, Student study mode, Assessment type and Absences

The attributes and the domain values for the selected attributes for the current study are defined in Table 2 for reference. A total of 129 student records (instances) for the year 2015/16 are involved in this investigation to develop the predictive model for the prediction of the students at high risk of failure in some of year 2 modules as the following:

- Algorithms and their Applications
- Usability Engineering
- Software Development and Management
- Year 2 Group Project

The predicted class attribute is **Overall Grade**, which is the final grade obtained by the student in the targeted module. It has five possible values A: Excellent, B:very Good, C: Good, D: Acceptable and F: Unacceptable or Fail, which have been merged later on to Low risk, Medium risk and High risk of failure to improve the classification results as explained in pre-processing section (see Table 3).

Table 2. Attributes of the students

	Attribute	**Description**	**Domain Values**
Category 1: Admission Data	Enrolment Status	Students enrolment status	{EE}
	Programme Name	Student program name	{UG Computer Science}
	Route Name	The student chosen route	{Computer Science, Computer Science (Artificial Intelligence), Computer Science (Software Engineering), Computer Science (Digital Media And Games), Computer Science (Network Computing)}
	Route Code	The code of the student chosen route	Based on Rout Code at the University
	Through Clearing	Whether the student enrolled in the same course as the course she/he has applied for	{Y, N}
	Fee Status	Tuition fee status	{Home/EU, Overseas}
	Student MOA	Students study mode	{FT, FSK, FT120, PT80, PT20}
	Detailed Fee Status	Tuition fee status	{Home, European, Overseas}
	Fee	The amount of paid fees	Based on the amount of paid fees
	Gender	Student gender	{M, F}
	Country of Domicile	Student country	Based on Student country
	Age on Entry	The student's age when he/she enrolled at the university	Based on Student age
	Qualification on Entry	Students previous qualification	{Foundation degree, Foundation course at level J, Higher education (HE) access course, A/AS level, Level 3 quals, all are subject to UCAS Tariff, Other qualification at level 2, International Baccalaureate (IB) Diploma, Non-UK first degree}
	CRS Code indicates	Payment method for	{Y, N}

	LBIC	the course	
	Location of Study	Campus name	Based on Campus name
	Admissions - Core Grades Flag	Indicates admissions decision for registering the student in the course	{Achieved, Predicted}
	Previous Institution	Student previous school or institution	{UK State School, UK Independent School, Any Non-UK Institution, UK Higher Education Institution}
Category 2: Level 1 (1st Year) Final Grades	Information Systems and Organisations_Grade	Module Final Grade	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
	Logic and Computation_Grade	Module Final Grade	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
	Level 1 Group Project Reflection_Grade	Module Final Grade	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
	Data and Information Assessment_Grade	Module Final Grade	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
	Software Design_Grade	Module Final Grade	{ A – Excellent, B – very Good, C - Good, D - Acceptable, F – Unacceptable}
	Software Implementation Event_Grade	Module Final Grade	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
	Fundamental Programming Assessment_Grade	Module Final Grade	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
Category 3: Module-Related Data	Course MOA	Module teaching mode	{FT, FSK}
	Tutor 1 Code	The code of the tutor at the university	Based on tutor code
	Tutor 1	The name of the tutor of the Module	Based on tutor name
	Module	Module Code at Brunel University	Based on module code in the university
	MAB_SEQ	Assessment code	{1, 2}
	MAB_NAME	Assessment type	{Unseen Examination, Assessment, Post-Mortem Style Group Review, Assessment of ethical and professional behaviour, Open book in-class Programming Test, Group submission of a design document plusprototype, Individual viva voce, Programming Assignment, Coursework (Practical Assignment) }
	MOA	Student study mode	{full time, part time}

	Supervisor	Student supervisor name	Based on supervisor name
	Absences	The total number of absences during the semester	Based on Module attendance count
Class Attribute	Overall Grade	Student overall grade in the Module	{ A – Excellent, B - very Good, C - Good,D - Acceptable, F – Unacceptable}
Merged Class Attribute	Overall Grade	Student overall grade in the Module after merging	{ Low risk – A and B , Medium risk - C ,High risk - D and F }

We performed steps for the implementation of the classification and clustering algorithms to predict the academic performance of the students for some of year 2 computer science core modules which are: Algorithms and their Applications, Usability Engineering, Software Development and Management and Year 2 Group Project using Java API and Weka Mining tool.

Since the number of students final grades classes is large with five possible values (A, B, C, D and F) and that will influence the performance of the predictive models, we merged students overall grades to reduce the number of classes for the targeted Modules using ‘Merges many values’ filter in Weka into three classes which are low risk, medium risk and high risk of failure classes. Low risk class is for students who have obtained A and B in the targeted module. Medium risk class is for students obtained C in the module. Whereas, the high risk class for students obtained D and F (see Table 3).

Table 3. Class Attribute regarding to student final grades

Class	**Grade Band**
Low risk	A, B
Medium risk	C
High risk	D, F

3.2. Clustering

Clustering is identifying groups of objects in which the objects of such groups are similar to one another in some aspects and different from the objects in the other groups [14]. Clustering is considered as the most applied unsupervised learning technique in data mining.

In educational data mining, clustering is applied to group the students according to their performance in the course into weak and strong students to help the weak students improve their studies [15] and [16]. Also, it used to identify the active and the non-active students based on their performance in course activities [5].

In our study we applied the simple K-Means clustering algorithm to each module using Java API in order to find interesting groups of student according to their final results (academic performance) in the predicted module. We obtained a number of three clusters which are cluster0, cluster1 and cluster2 providing adequate correlations of student groups with the class attribute Overall Grade (academic performance).

3.3. Classification

Classification, a form of supervised learning, is a very common data mining technique that is applied to map datasets into sets of classes [5]. To develop such models, the data undergo a process that consists of learning and classification. In the learning process, the training set is analysed using classification algorithms to generate logical rules based on the relation between the selected attributes. Consequently, the classification process identifies the accuracy of the model by applying obtained rules on the test sets to evaluate the classifier [13].

The machine learning algorithms applied for classification process in this study were naïve Bayes and C4.5 decision tree. Since the dataset was not large with only 129 student records, we encountered class implanting issue. To solve this, we applied the Synthetic Minority Oversampling Technique (SMOTE) to the minority class which was the Medium risk class in order to resample the dataset. When applying this technique, new minority class instances are created based on the percentage of SMOTE for the minority class.

We obtained the test results of the predictive models by10-fold cross validation evaluation method. The predictive models 'resulted from the classification process' illustrate ways to identify whether the student at high, medium or low risk of failure.

4. EXPERIMENTAL RESULTS

The student datasets used in this study WAS analysed using Java API and Weka Mining tool with two classification algorithms used to develop the predictive models, those were Naïve Bayes and C4.5 Decision tree. A comparison of accuracy of the selected classification algorithms is provided in Table 4 and Figure 1. In fact, Algorithms and their Applications Module obtained the highest accuracy result in both Naïve Bayes and C4.5 decision tree (see Table 4) comparing to other Modules. However, all the predictive models produced accurate results in terms of (69%-84%) compared to what found in the literature.

Table 4. Accuracy Comparison of predictive models

Module title	**Naïve Bayes Accuracy**	**J48 Decision Accuracy**
Algorithms and their Applications	88.48%	84.29%
Usability Engineering	70.31%	70.31%
Software Development and Management	69.11%	75.39%
Year 2 Group Project	87.33%	84.16%

The sensitivity analysis of the predictive models summarised in Table 5 illustrates the comparison of True Positive rate (TP) and the False Positive rate (FP) of the applied algorithms (Naïve Bayes and C4.5 Decision tree) on different modules. The highlighted probabilities in the following table indicate the highest TP rates and the lowest FP rates were found at high risk failure for each specific module. In particular, the probability of correctly detection of high risk failure in "Algorithms and their application" module is identified by the highest TP rate of 0.969 and 0.938 exploiting Naïve Bayes and C4.5 Decision tree, respectively.

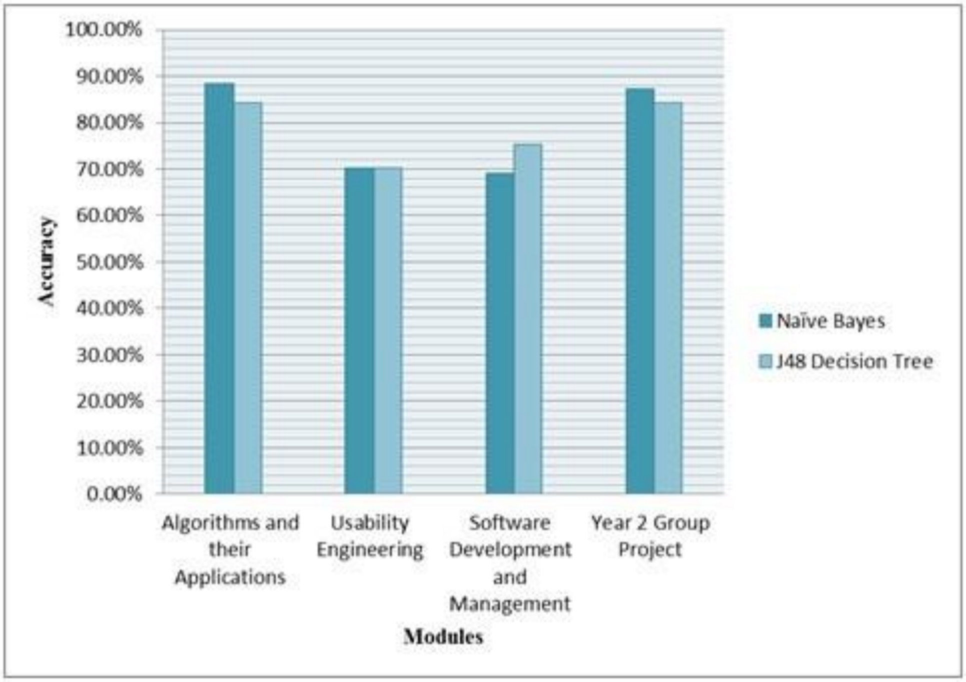


Figure 1. Accuracy Comparison of the predictive models

Table 5. TP Rate and FP Rate Comparison of the predicted modules

Module Title	**Class**	**Naïve Bayes** **TP Rate**	**FP Rate**	**C4.5 Decision Tree** **TP Rate**	**FP Rate**
Algorithms and their Applications	low risk	0.821	0.081	0.776	0.121
	medium risk	0.867	0.076	0.817	0.076
	high risk	**0.969**	**0.016**	**0.938**	**0.039**
Usability Engineering	low risk	0.671	0.205	0.714	0.213
	medium risk	0.192	0.066	0.192	0.066
	high risk	**0.865**	**0.219**	**0.833**	**0.208**
Software Development and Management	low risk	0.806	0.327	0.921	0.500
	medium risk	0.379	0.160	0.517	0.117
	high risk	**0.391**	**0.095**	**0.043**	**0.012**
Year 2 Group Project	low risk	0.732	0.072	0.610	0.056
	medium risk	0.882	0.039	0.882	0.059
	high risk	**0.920**	**0.083**	**0.902**	**0.147**

Figure 2 presents the best preform C4.5 decision tree model that predicts the students at high risk of failure. Student Overall Grade is the predicted feature in this classification model, and only a number of features were considered (8 of 33). Interestingly, remarkable result to emerge from the predictive model is that, student qualification has a high impact on the prediction of the high risk of failure students. Furthermore, some of level1 Modules final grades are highly influencing the prediction result. These Modules are Information Systems and Organisations Module, Logic and Computation Module and Software Implementation Event Module which are the core Modules of year 1 of computer science program.

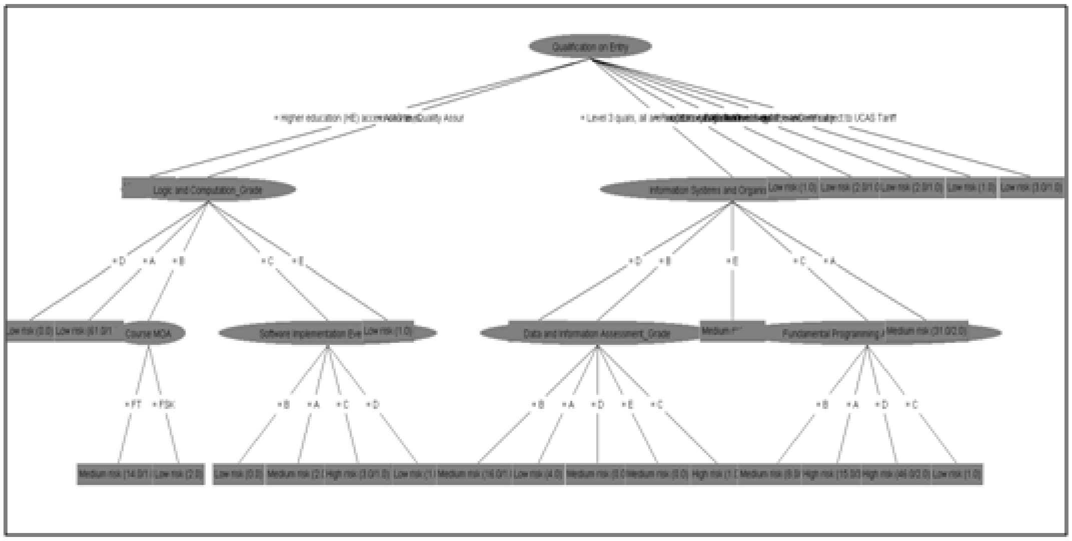


Figure 2. Algorithms and their Applications C4.5 Decision Tree Output

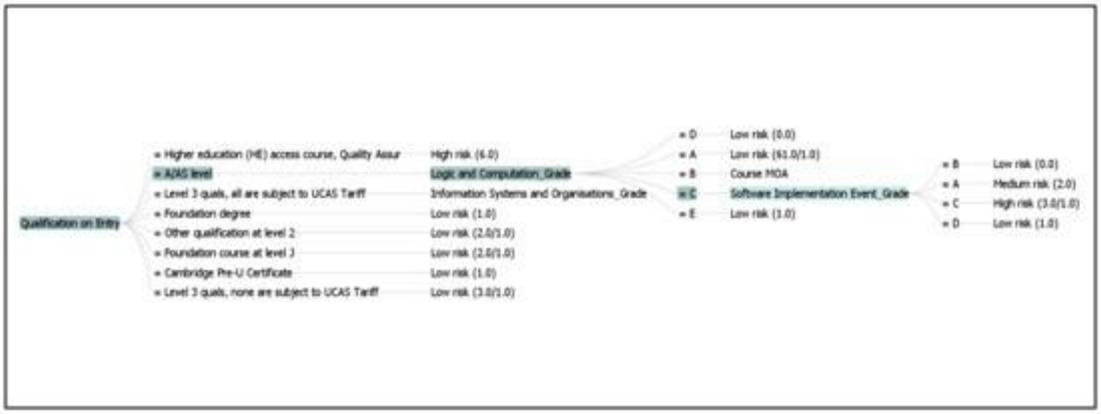

Qualification on Entry
= Higher education (HE) access course, Quality Assur
= A/AS level
= Level 3 quals, all are subject to UCAS Tariff
= Foundation degree
= Other qualification at level 2
= Foundation course at level 3
= Cambridge Pre-U Certificate
= Level 3 quals, none are subject to UCAS Tariff
High risk (6.0)
Logic and Computation_Grade
Information Systems and Organisations_Grade
Low risk (1.0)
Low risk (2.0/1.0)
Low risk (2.0/1.0)
Low risk (1.0)
Low risk (3.0/1.0)
= D
= A
= B
= C
= E
Low risk (0.0)
Low risk (61.0/1.0)
Course MOA
Software Implementation Event_Grade
Low risk (1.0)
= B
= A
= C
= D
Low risk (0.0)
Medium risk (2.0)
High risk (3.0/1.0)
Low risk (1.0)

Figure 3. Algorithms and their Applications C4.5 Prefuse Tree Output

From the Prefuse tree in Figure 3 'which is Weka visualization tool that uses Prefuse toolkit to best explore the generated tree' we can extract some interesting rules that ended to high risk students. These rules indicate the influence of student's qualification on their academic performance in Algorithms and their Applications Module, for example:

1. **if** Qualification on entry **=** Higher Education (HE) access course **then** high risk**;**

2. **if** Qualification on entry **=** A/AS Level **ʌ**Logic and Computation_Grade = C **ʌ** Software Implementation Event _Grade = C **then** high risk**;**

5. CONCLUSION AND FUTURE WORK

This study is an attempt to apply C4.5 and Naïve Bayes classification methods to analyse level 2 students' academic performance based on their admission, course related data and level 1 final grades. The main goal of the current investigation was to develop a predictive data mining model for students' academic performance in university level so to identify the high risk of failure students. The second aim was to identify the key features affecting the predictive model.

By applying C4.5 and Naïve Bayes algorithms we revealed that Naïve Bayes performs better than C4.5 decision tree algorithm in predicting the students at high risk of failing the Module with an accuracy result of 88.48% for Naïve Bayes and 84.29% for C4.5 algorithm. Another major finding was that student qualifications on entry have high impact on students' academic performance. Moreover, some of level1 Modules final grades are influencing the results of the students in level2 Modules.

These findings provide the following insights for future investigation in Education Data Mining. The prediction of students' performance could be influenced by other factors or features. We are attempting to investigate other student's features that may influence the prediction process and provide better accuracy results. Moreover, different classification algorithms could be applied to obtain better predictive models using the same dataset.

ACKNOWLEDGEMENT

We would like to thank Dr Simon Kent, the director of education in the Department of Computer Science at Brunel University London for his professional cooperation throughout this study. Also, special thanks to Ms. Sara Brown, the student programmes manager at the department for her assistant in providing students datasets.

REFERENCES

[1] Hand, D. J., Mannila, H., and Smyth, P. (2001). Principles of Data Mining.MIT Press.

[2] Baker, R. (2010) Data Mining for Education. In McGaw, B., Peterson, P. and Baker, E. (Eds.) International Encyclopaedia of Education (3rd edition), vol. 7, pp. 112-118.Elsevier, Oxford, UK.

[3] Cortez, P., Silva, A., (2008) Using data mining to predict secondary school student performance. Presented at the 5th Annual Future Business Technology Conference, EUROSIS, pp. 5–12.

[4] Al-Radaideh, Q., Al-Shawakfa, E., Al-Najjar, M., (2006) Mining Student Data Using Decision Trees (PDF Download Available). Presented at the International Arab Conference on Information Technology (ACIT'2006), Jordan.

[5] Aher, S., L.M.R.J., L., (2011) Data Mining in Educational System using WEKA. Presented at the International Conference on Emerging Technology Trends (ICETT), International Journal of Computer Applications® (IJCA), pp. 20–25.

[6] Yadav, S., Pal, S., (2012) Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification. World of Computer Science and Information Technology Journal (WCSIT) 2, 51–56.

[7] López, M.I., Luna, J.., Romero, C., Ventura, S., (2012) Classification via clustering for predicting final marks based on student participation in forums. Presented at the The 5th International Conference on Educational Data Mining, ERIC, pp. 148–151.

[8] Mayilvaganan, M. and Kalpanadevi, D., (2014) Comparison of Classification Techniques for predicting the performance of Students Academic Environment in: 2014 IEEE Conference on Communication and Network Technology (ICCNT), pp. 113-118

[9] Kabakchieva, D., (2013) Predicting Student Performance by Using Data Mining Methods for Classification. Cybern. Inf. Technol. 13, 61–72. doi:10.2478/cait-2013-0006

[10] Kaur, G., Singh, W., (2016) Prediction Of Student Performance Using Weka Tool. Vidya 17, 8–16.

[11] Kaur, P., Singh, M., Josan, G., (2015) Classification and prediction based data mining algorithms to predict slow learners in education sector. Presented at the 3rd International Conference on Recent Trends in Computing 2015(ICRTC-2015), Elsevier B.V, pp. 500–508.

[12] Bekele, R., Menzel, W., (2005) A BAYESIAN APPROACH TO PREDICT PERFORMANCE OF A STUDENT (BAPPS): A Case with Ethiopian Students. Presented at the IASTED International Conference on Artificial Intelligence and Applications, part of the 23rd Multi-Conference on Applied Informatics, Innsbruck, Austria.

[13] Bhardwaj, B.K., Pal, S., (2012) Data Mining: A prediction for performance improvement using classification. ArXiv12013418 Cs.

[14] El-Halees, A., (2009) Mining Students Data to Analyze Learning Behavior: A Case Study (PDF Download Available). Dep. Comput. Sci. Islam. Univ. Gaza PO Box 108.

[15] Hogo, M.A., (2010) Evaluation of e-learning systems based on fuzzy clustering models and statistical tools. Expert Syst. Appl. 37, 6891–6903. doi:10.1016/j.eswa.2010.03.032

[16] Perera, D., Kay, J., Koprinska, I., Yacef, K., Zaiane, O., (2009) Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. IEEE transactions on knowledge and data engineering 21, 759–772. doi:10.1109/TKDE.2008.138

AUTHORS

Mashael Al-luhaybi is a Lecturer in eLearning and Distance Education at Umm Al-Qura University, Saudi Arabia. She is currently a PhD candidate in Machine learning in particular Educational Data Mining at Brunel University London, UK. She obtained her MSc from the University of Brighton, UK in 2011.She is interested in predicting student academic performance and detecting their learning behaviour.

Allan Tucker is a Senior Lecturer at Brunel University London, United Kingdom. He is the Head of Intelligent Data Analytics (IDA) Research Group at Brunel University. His research interests lie in modelling of brain function, human and animal behaviour. He obtained his PhD from Birkbeck, University of London.

Leila Yousefi is a PhD candidate in Machine Learning at Brunel University London, UK. She obtained her MSc from Azad University of Qazvin, Iran. She is a member of the Intelligent Data Analytics (IDA) Research Group at Brunel University London. Her research interest is in Artificial Intelligence in Medicine and Data Mining.

DEEP LEARNING BASED DATA GOVERNANCE FOR CHINESE ELECTRONIC HEALTH RECORD ANALYSIS

Junmei Zhong[1], Xiu Yi[2], Jian Wang[2], Zhuquan Shao[2], Panpan Wang[2], and Sen Lin[2]

[1]Inspur USA Inc
2010 156th Ave NE Bellevue, WA 98052
[2]Inspur Software Group, Technology Center
1036 Langchao Rd., Jinan, China

ABSTRACT

Electronic health record (EHR) analysis can leverage great insights for improving the quality of human health care. However, the low data quality problems of missing values, inconsistency, and errors in the data columns hinder building robust machine learning models for data analysis. In this paper, we develop a methodology of artificial intelligence (AI)-based data governance to predict the missing values or verify if the existing values are correct and what they should be when they are wrong. We demonstrate the performance of this methodology through a case study of patient gender prediction and verification. Experimental results show that the deep learning algorithm works very well according to the testing performance measured by the quantitative metric of F1-Score, and it outperforms support vector machine (SVM) models with different vector representations for documents.

KEYWORDS

EHR Analysis, Data Governance, Vector Space Model, Word Embeddings, Machine Learning, Convolutional Neural Networks.

1. INTRODUCTION

Electronic health record (EHR) analysis can leverage great insights for improving the quality of human health care and it is one of the approaches to accomplishing precision medicine. However, there are a lot of challenges in analyzing such massive data set. One of the most challenging problems for our big Chinese EHR data analysis is its low data quality. The typical issues include missing values, inconsistency, and errors in the data columns, which hinder building robust machine learning models for data analysis. Since for such massive data set it is impossible to do data correction in a manual way, it is very desirable to develop some automatic algorithms which can make corrections and verifications for the individual problems in the big data. In this paper, we develop an artificial intelligence (AI)-based data governance strategy

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 91–103, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80507

trying to leverage the power of AI algorithms in analyzing big data of patient clinic profiles to predict the missing values or verify if the existing values are correct and what they should be when they are wrong. Particularly we develop natural language processing (NLP), traditional machine learning and deep learning techniques for gender prediction and verification as a case study. Although it is only used for patient gender prediction here, the underlying fundamental principles of this methodology can be applied for the prediction of other kinds of missing values and verification of other kinds of existing values. Also, gender prediction from modeling user behavior profile plays a significant role in targeting and personalized product recommendation in the e-Commerce of digital advertising. For EHR analysis, patients' gender information plays a very important role in referring some useful information. However, in the Chinese EHRs we obtained, the gender information in many cases is either missed or not correct. Even if we can use the patients' Chinese identity (ID) information to extract the gender information, however, for privacy consideration, the ID information is usually hidden and not available for data analysis tasks. So, we need to develop an effective solution without using the patients' ID information. We also agree the point that it is possible to guess the patient's gender information with a pretty high accuracy from his/her name without resorting to the sophisticated AI algorithms, but we would like to emphasize that we are trying to develop a general methodology for data governance with AI algorithms for predicting all missing-values or verifying all existing values, not only for the specific gender prediction problem. For example, we are using this strategy to predict the category of each patient's diseases according to the Chinese national medical coding standard.

Through information fusion, we first construct each patient's clinic profile from the heterogeneous tables of symptom description, medical treatment process, lab tests, together with the doctor's prescription data, and take it as a document, trying to leverage some insights from such kind of clinic information by using supervised machine learning to predict and verify the patient's gender information. There are two basic components in the system. The first one is NLP for tokenization and the feature engineering for document's vector representation, and the second one is supervised machine learning with both traditional machine learning and/or deep learning algorithms. When using machine learning, we need to get the document's vector representation and we have tried 4 different representations: the bag of words (BOW) vectors with both TF-IDF weighting and binary representation for tokens, the averaged word embeddings of words in the document pretrained with theword2Vec [2] algorithm, and the document vector obtained with the doc2Vec [3] algorithm. For supervised machine learning algorithms, we have tried the multi-class support vector machine (SVM) [4, 5], one of the most efficient traditional machine learning algorithms for classification, and the convolutional neural networks (CNN) [6] of deep learning. For multi-class SVM, we have tried different vector representations for documents as the feature vectors for SVM, trying to get its best classification performance. Experimental results show that the CNN with inputs of word embeddings works best according to the testing performance based on the quantitative metric of F1-Score. It outperforms support vector machine (SVM) with different document vector representations. To our best knowledge, this is the first work in using NLP and deep learning for data governance in EHR analysis, especially for patient gender prediction from clinic profile.

Our contributions are as follows:

- We develop AI-based data governance strategy for EHRs. The AI algorithms include NLP, machine learning and deep learning algorithms.

- We construct patients' clinic profiles by using information fusion methodology from heterogeneous tables and records in the EHRs according to the domain knowledge of medical informatics for gender prediction. This makes it possible to leverage insights from the big EHR data using AI algorithms. The success of this methodology can be applied to the prediction of other missing values.

The rest of the paper is organized as follows. In Section 2, we discuss the methodology of feature extraction and vector representations for documents. Section 3 talks about training the SVM mode of traditional machine learning algorithms, and CNN of deep learning in text classification. In Section 4, we present the experimental results. We conclude the paper with discussions and future work in section 5.

2. METHODOLOGY

The system of AI-based data governance for gender prediction and verification consists of 3 components. Data preparation through information fusion, vector representation for documents with NLP algorithms and word embeddings for tokens, and traditional machine learning and deep learning algorithms for gender prediction and verification.

2.1 Data Preparation Through Information Fusion

The Chinese EHRs we are using for our projects consist of 179 heterogeneous tables. However, since creating EHRs is still at the beginning stage in China and in most of the time, doctors are more willing to write the paper notes for their patients rather than leave electronic notes in the computer system. So, in the 179 tables, many of them do not have much useful information for the empty columns in the tables. After preprocessing, we get 85 tables, but even in these 85 tables, many records still have missing values and we need to do additional filtering process to remove the useless records according to our application. We analyze these tables with the domain knowledge of medical informatics to construct individual complete and meaningful clinic profiles about each patient's individual symptom descriptions, lab tests, doctor's treatment plans, and prescriptions for medications. Then we concatenate the text data in a few columns from each patient's clinical profile and take the concatenated text data as a document for NLP and machine learning. Our motivation is to make full use of the massive clinic data for AI algorithms to predict the patients' gender information.

2.2 Vector Representation for Documents

Machine learning algorithms take individual documents as the inputs, so we first tokenize each document into a collection of terms. For this, we use Han LP[1], an open source software, for the Chinese document tokenization. Since there are no spaces between the Chinese words in each sentence, which is totally different from the English texts which can be separated by spaces between the individual words, the tokenization of Chinese texts needs specific algorithms and it is another hot research topic, here we only use the available tool for this task and focus our efforts on other things based on the tokenization result. Then we generate a vector representation for each document with different ways, which include the TF-IDF weighting method for tokens, binary representation for tokens, the averaged word embeddings of word2Vec[2],doc2Vec [3]for individual documents, and the vectors of words as the inputs of CNN [6] to get the vector representation for documents.

2.2.1 The Bag of Words (BOW) Method

The BOW method makes use of tokenized individual words and/or N-Grams (N>=1)in a corpus as features for a document's vector representation, which is usually called a feature vector in machine learning and pattern recognition. All N-Grams constitute the vocabulary of the corpus. If N is equal to 1, the N-Gram is called unigram. For individual tokens, we usually have both binary representation and TF-IDF weighting representation to get the feature values. The binary representation does not count the number of occurrences of the tokens but only considers their presence and absence in the individual documents. If a token is present in the document, the vector's corresponding feature value is 1, otherwise 0. On the other hand, the TF-IDF weighting method takes the product of two statistics, the term frequency and inverse document frequency. The term frequency is simply the number of times that the term t appears in a document d. It assumes that the more frequent a token appears in the document, the more important it is for the topics of the document and it is usually calculated in the following augmented way [8]:

$$t\dagger(t, d) = 0.5 + 0.5 * \frac{\dagger^{t,d}}{\max\{\dagger_{t^{F},d} : t^{F} \in d\}} \quad (1)$$

where $\dagger_{t,d}$ denotes the frequency of term *t* in document *d*.At the same time, the inverse document frequency is calculated in the following way [8]:

$$idf(t, D) = \log(\frac{N}{\dagger\{d \in D : t \in d\}\dagger} + 1) \quad (2)$$

With

- N the total number of documents in the corpus D, N= |D|
- The denominator |{d ∈ D: t ∈ d}| is the number of documents where the term *t* appears. If the term *t* does not occur in the corpus, the denominator needs to be adjusted into |{d∈ D: t ∈ d}|+1

The inverse document frequency is used to offset the impact of common words without having specialty. But the BOW method usually suffers from the following issues:

- Sparsity, most of the documents usually have only a small fraction of the vocabulary, and most of the words in the vocabulary are absent from individual documents, resulting in the term-document matrix with a lot of unwanted zeros.
- Does not take the word order information into account and only considers the occurrence of a word independent of the others, which is not true from both semantic and syntactic point of view. So, documents with different semantic meanings may be taken to be similar only if they contain the same words.

High dimensionality. Corpora generally have at least thousands of words (features). In addition to this, if the 2-grams and 3-grams are included, the number of features per document increases significantly. It could generate an even more sparse term-document matrix leading to insufficient RAM problem when we try to hold the entire matrix into the RAM. Not all features are important and modeling the data with such features needs a huge number of annotated samples for training, and it tends to lead to the overfitting problem for supervised learning when no sufficient annotated samples are provided. The high dimensionality of data challenges very much supervised machine learning algorithms for the curse of dimensionality, and in most of the time, we need to do dimensionality reduction for the BOW vector representations. But it is very critical in dimensionality reduction to preserve the structural information of the data

2.2.2 Word2Vec

Since the above BOW-based vector representation is not efficient to capture the semantic information from documents with the limitations of high dimensionality and sparsity, researchers have proposed different methods to represent documents and words in an embedded low-dimensional continuous vector space and the word2Vec [2] generates the state-of-the-art results. The Word2vec algorithm is such a distributed representation learning method to extract both semantic and syntactic information for individual words in a sentence. It consists of a bunch of related models that are used to produce the distributed representation of word embeddings. These models are the continuous bag-of-words (CBOW) and the skip-gram as shown in Figure 1. The CBOW model predicts the current word from its surrounding context words within a window centered at the current word, while for the skip-gram model, given the current word, it predicts the surrounding context words within a window for this current word. The Word2vec model is an unsupervised learning algorithm which can be trained with the hierarchical softmax and/or negative sampling method to reduce the computational complexity and make the learning process practical. In the hierarchical softmax method, a binary Huffman tree is constructed for the terms in a corpus according to their frequencies to reduce the computational complexity in updating the vectors of the terms. The benefits of using the Huffman tree is that all words are the leaf nodes in the tree, and high frequency words will have short paths from the root of the tree to such leaf nodes and low frequency words will have long paths. In the iterative backpropagation process, for updating each word's vector, we do not need to update the vectors of all other words in the vocabulary, but only need to update the vectors of the nodes in the path from the root of the tree to the leaf node in the tree. If the word is a frequent word, its corresponding path will be short and the further reduction of the computation complexity is accomplished. On the other hand, for the negative sampling method, it accomplishes this goal and improves the vector quality of low-frequency words byonly sampling a few negative samples from the vocabulary for updating their vectors. For this end,in the negative sampling method, the high-frequency words are down-sampled and the low-frequency words are up-sampled by lifting the low-frequencies.

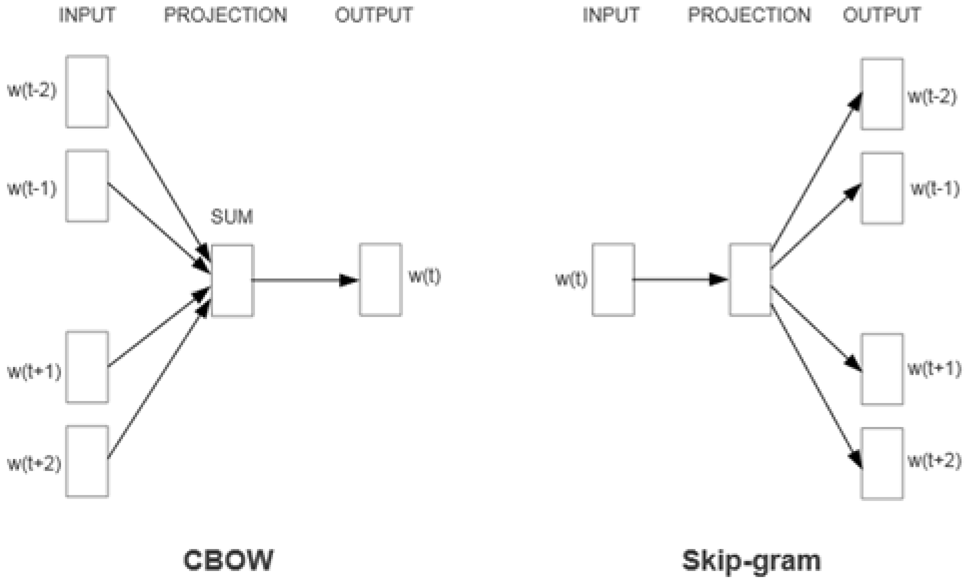


Figure 1. The illustration of CBOW and Skip-gram models in Word2Vec with courtesy of Mikolov etc. [2]

These models are the two-layer shallow neural networks. Word2vec takes as its inputs the high dimensional one-hot vectors of the words in the corpus and produces a vector space of several hundred dimensions which are much smaller than the size of the vocabulary, such that each unique word in the corpus is represented by a continuous dense vector in the embedded vector space. A very salient feature of this kind of vector representation with the word embeddings is that word vectors are such points in the vector space that for semantically similar words, their vectors are close to each other. This offers great benefit that we can infer the semantically similar words from the vector space if one word's vector is known, and it hence has been attracted with tremendous attention in text analysis. However, the word embeddings still have a limitation for representing documents. The usual way of using the word vectors is to take the averaged word embeddings of words in a document for document classification, sentiment analysis for movie reviews and customer service reviews, and text clustering analysis, but for most of the documents composed of syntactic sentences, this kind of average operation will introduce noise to the average result, making it deviate from the actual topic of the document.

2.2.3 Doc2Vec

The Doc2Vec algorithm [3] is an extension of Word2Vec for representing a document or a paragraph with a single unique real-valued dense vector learned together with the process of generating the vectors for individual words in the corpus. This is accomplished by assigning a unique document tag to each document, and the vector of this added document tag is learned together with all other words in the same document. When the learning process is done, the vector of the document tag is obtained and it is used for document analysis. Also, this model can be used to infer the document vectors for new documents. Just like word vectors generated by word2Vec, which provide semantic inference for individual words, a document vector generated by doc2Vec as shown in Figure 2, can be thought of reflecting some semantic and topic information of the document. As a result, the document vectors of similar documents tend to be close to each other in the vector space and they are very useful for document classification or clustering analysis.

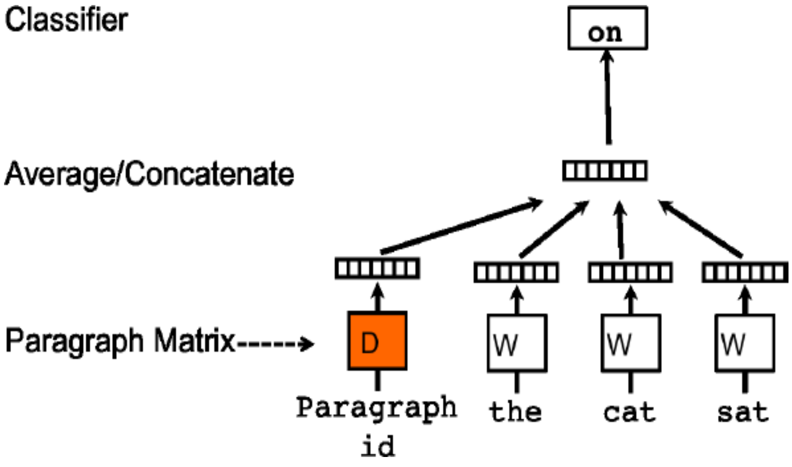


Figure 2. A framework for learning paragraph vector with courtesy of Quoc Le, etc. [3]

2.2.4 The CNN Architecture for Text Classification

CNN is one of the deep learning algorithms and it integrates feature extraction, feature selection and classification in a single architecture. It has been widely used for computer vision [7] and text classification [6]. As outlined by Figure 3, for computer vision, different channels of the image, like the R, G, B colors of the image, can be used as the inputs of CNN architecture for convolutional feature extraction. For text classification, the CNN usually takes as inputs the word embeddings of the sentence by stacking the words' vectors as a matrix according to the order of the words in the sentence. The embeddings can be either from word2Vec, one-hot representation or other vector representations of words in a sentence, forming different channels for representing the text data. With the CNN architecture, each channel of the texts can be represented as a matrix, in which, the rows represent the sequence of tokens or words in a sentence, and each row is a word's embeddings. The matrix is convolved with some filters of different sizes such as 3, 4, 5, but with the same dimension as the words' embedding vectors.

The main idea of CNN for text classification with different sizes of filters is to extract the semantic features of the N-Grams with the filters. The different filter sizes correspond to different numbers of the N in N-Gram. The words' vectors can be either from the pre-trained word embeddings such as those of word2Vec from large corpus, or randomly initialized. For the latter case, the word vectors are iteratively updated by backpropagation during the training process until the model is learned and the word vectors become the side effects of the CNN. Let's assume the filter size is *m*, sentence length is *l*, dimensionality of word embeddings is *d*, then the sentence matrix $x \in R^{S\times d}$and the filter can be represented as a matrix $w \in R^{N\times d}$. During the convolution process, each of the filters gradually moves down one word at a time along the sequence of words and at each position, the filter covers a portion of the words' vector matrix, i.e., m words' vectors in the matrix, and the point wise multiplication of the filter with the covered vector matrix is taken, and the multiplication results are summed up. This sum is then taken by the rectified linear unit (Relu) activation function together with a biased term $b \in R$to generate a feature value:

$$c_i = \mathbf{f}(w \cdot x_{i:i+N-1} + b) \tag{3}$$

After the convolution process is done, a list of feature values is obtained like$c = [c_1, c_2, c_3, \ldots, c_{S-N+1}]$, which is regarded as the feature map of the filter. Finally,the max-pooling operation continues to take the maximum value from the feature map as the feature value of the filter's convolution result with the sentence. When all filters are applied for convolution with the sentence's vector matrix, we can get a list of feature values as the feature vector representation for the input sentence data. This feature extraction process with the max-pooling operation makes the length of the final feature vector independent of the input sentence length and the filter sizes.The length of the final feature vector is only dependent on the number of filters used for convolution. The final step in the CNN architecture is a full connection including the dropout strategy, regularization, and the Relu activation function from the final feature vectorwith the output layer and this full connection layer is fundamentally like the conventional neural network. The classification result of a sample is obtained by the softmax function applied to the output layer. The number of neurons in the output layer depends on the number of classes for the output.

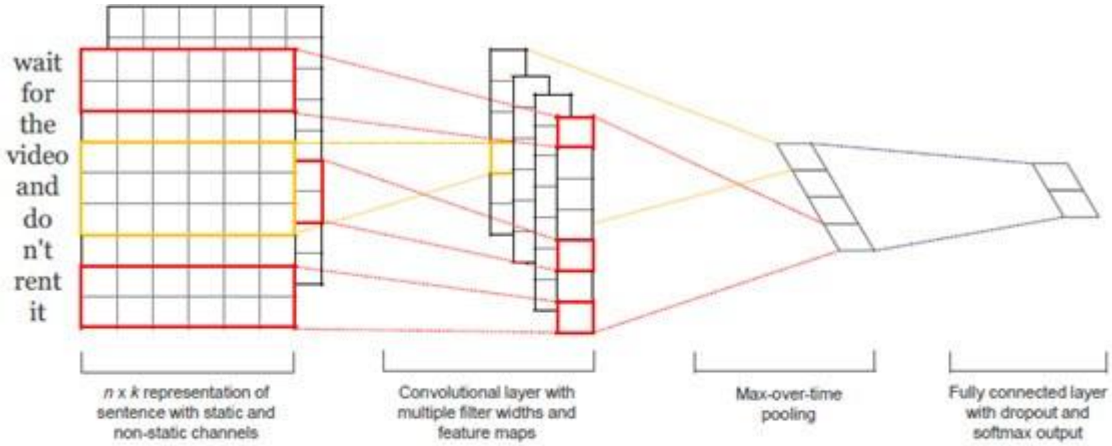

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3. The CNN architecture for text classification with courtesy of Yoon Kim [6].

3. TRAIN MACHINE LEARNING MODELS

We use machine learning algorithms to model patients' clinical profile for gender prediction. In the EHRs, many patients' clinical profiles do not have clear gender implications, it is very hard to predict the patients to be male to female. As a result, we add one more class of "unknown" in addition to the two classes of "male" and "female", forming a 3-class classification problem.

3.1 Training the CNN Model

The CNN is used for a 3-class classification problem. The Tensorflow's Python implementation of the CNN is used in this work. We use the pretrained word embeddings of word2Vec as the inputs of CNN. We use 100 filters for each filter size for feature extraction in the convolutional process. The corpus for training the word2Vec algorithm to get the word embeddings is obtained from the crawled Chinese medical documents. For the hyperparameters of CNN, through grid search, we set batch size 20, epochs size 40, the dimension of word embeddings200, dropout 0.5, andl_2-norm is used as the regularization. Additionally, a threshold is set 3 for clipping the gradient magnitude. We shuffle the samples in each epoch.

3.2 Training Multi-Class SVM

For performance comparison, the traditional machine learning algorithm of multi-class SVM is used as the baseline model. Although SVM belongs to the traditional machine learning models, it is very different from many of the other models like artificial neural network, decision tree, and Bayesian models for which the goal is to minimize the empirical learning risk, represented by the mean squared error of the training samples with respect to their predictions by the corresponding model. But for SVM, its goal is to minimize the structural risk by maximizing the margin between the two classes of the training samples. Minimizing the empirical learning risk does not have the guarantee for the model to generalize well to the unseen samples, but as shown in Figure 4, minimizing the structural risk by maximizing the margin between the two classes can guarantee for SVM to generalize well to the unseen samples, and this is what we are pursuing in training a machine learning model.

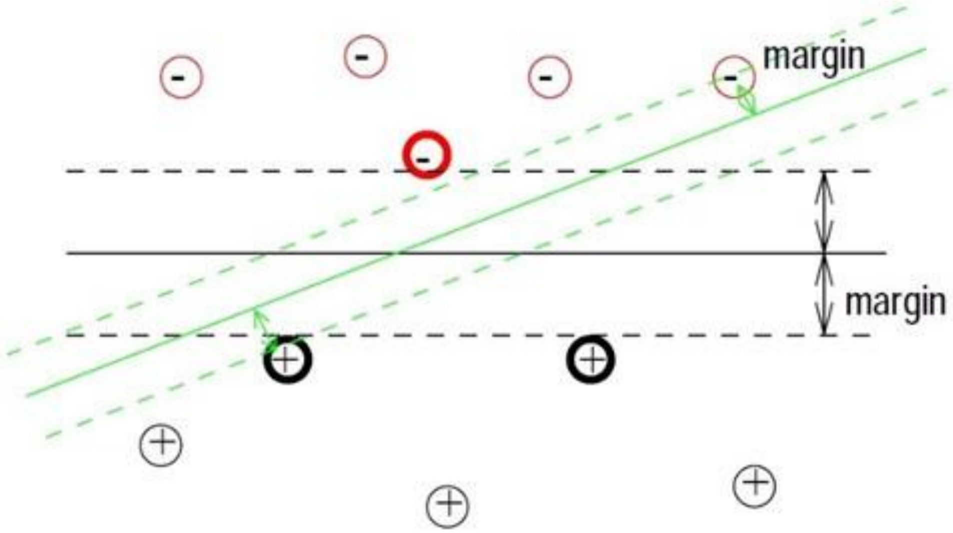


Figure 4. The illustration of the maximum margin for SVM

When training machine learning models, we do not want our model to work only well on the training data, but we want the model to be able to work well on the unseen data in the future, so the SVM model matches our goal of training a machine learning model. In Figure 4, we have two classes of samples denoted by empty red circles, and black circles, respectively, for classification and there are two possible margins for the hyperplane to be placed to separate the two classes of samples, one is labeled with the green color and the other is labeled with the black color. We can clearly see that the hyperplane marked with the black color can generalize better than the one with the green color because it has a wider margin. To train the SVM model, its input feature vectors of documents are from either doc2Vec, or the averaged word embeddings of all words in the document with word2Vec, or the TF-IDF vectors, or the binary vectors of tokens, respectively. Furthermore, in our experiment, only linear SVM is used and no kernel SVM is used for the fact that the dimensionality of the feature vector is already high. For the implementation of the multi-class SVM, we use the Python tool of Scikit-learn, which is a free software of machine learning library in Python programming language. We use the grid search method to optimize the hyperparameters of SVM.

4. EXPERIMENTAL RESULTS AND ANALYSIS

In the experiment, we compare CNN with SVM, one of the most popular traditional machine learning algorithms. Also, 4 different vector representations for documents are used as the input feature vectors for SVM. When obtaining the pre-trained word vectors with word2Vec, we have tried 4 models:

- CBOW + hierarchical softmax,
- CBOW + negative sampling,
- Skip-Gram +hierarchical softmax, and
- Skip-Gram + negative sampling.

For our dataset, the CBOW+negative sampling works best and it is selected to generate the word embeddings for CNN’s inputs, and the averaged word embeddings for the SVM. In this paper, the quantitative metrics used for measuring the performance of the models are the precision (accuracy), recall, and F1-score and they are calculated in the following way:

$$\text{Precision} = \frac{\text{tp}}{\text{tp}+\dagger\text{p}} \qquad (4)$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp}+\dagger\text{n}} \qquad (5)(5)$$

$$\text{F}_{1\_}\text{score} = 2 \cdot \frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \qquad (6)(6)$$

Where *tp* denotes true positives, *fp* denotes false positives, and *fn* denotes false negatives. The prediction results of CNN with word embeddings as inputs is listed in Table 1.

For training the SVM model, we have compared the binary BOW vector with the TF-IDF BOW vector. However, it is found out that the performance of SVM with the TF-IDF vector representation is far below that with the binary BOW vector representation, so its result is not listed in Table 2 and only the binary vector representation is used as the method of BOW vectors. Our analysis figures out that the vector representation based on TF-IDF weighting method calculated with formula (1) and (2) is suitable for long documents with many words because it introduces some additional “smoothing” effect, but for our short documents, the binary vector representation makes more sense to represent them. The experimental results with comparative studies show that the CNN works best and it outperforms the SVM model with different feature vectors for document representation. We think this is mainly for the following reasons. The vector representation of a document with the doc2Vec is trying to summarize the document’s topic, but for our documents, it cannot extract very much topic information because the documents are very short and they are composed of only a list of words about the symptoms of the disease from different aspects without containing any syntactic information. The fact that the averaged word embeddings ofword2Vec accomplishes better performance than the doc2Vec for SVM, is that the words in each document are related to each other about the disease, and to some extension, they can be regarded as the synonyms of each other in our EHR corpus, and their vector representations in the embedded vector space are close to each other. As a result, when each of our documents does not have any syntactic information and is only composed of such words, the averaged vector of these words’ vectors will still be close to the vectors of these words and it can be roughly taken to be the “center” of these words’ vectors. So, for our documents, the averaged word embeddings represents each document better than the document vector obtained with the doc2Vec algorithm. But since we still do not have very sufficient data to train the word2Vec algorithm for generating high-quality word embeddings, its performance with SVM is still lower than that of SVM with the binary BOW vector. It is reported that for generating the high-quality word embeddings with word2Vec [2], a very huge corpus of 100 billion words from Google news is used for training, so, their obtained word vectors can sufficiently capture the semantic information of words. Furthermore, for the deep learning structure CNN, it further extracts effective features from the input word embeddings for classification by using different sizes of filters to extract the N-Gram semantic information in the texts, nonlinear activation functions, max-pooling operations, dropout sampling for preventing the correlations in features, and an additional learning layer through the full connection with the output layer. As a result, even if the deep learning architecture in this paper is only the shallow CNN, it could accomplish the best performance among all these models. But for the multi-class SVM with different document's vector representations as inputs, it simply takes what is provided as the feature vector for learning, and there is no additional feature engineering work as done in CNN to re-extract and re-select the most effective features from the input features for classification.

Table 1. The prediction results of CNN with word embeddings as inputs.

Input Vectors	F1-Score	Accuracy	Recall
Word2Vec	0.96	0.94	0.97

Table 2. The prediction results of SVM with different feature vectors for document representation.

Input Vector(s)	F1-Score	Accuracy	Recall
Binary BOW	0.93	0.97	0.90
Avg. Word2Vec	0.91	0.97	0.86
Doc2Vec	0.66	0.60	0.74

5. CONCLUSION AND FUTURE WORK

In this paper, we develop NLP and supervised machine learning based data governance strategy for EHR analysis and demonstrate its effectiveness in gender prediction. Two kinds of machine learning algorithms are investigated. They are the traditional machine learning algorithm SVM and deep learning algorithm CNN. Four kinds of vector representations are investigated for document's vector representation for the multi-class SVM. From our experimental results, the deep learning CNN algorithm outperforms the state-of-the-art traditional machine learning algorithm SVM. This AI-based data governance strategy can be applied to any other data prediction and verification problem to improve the data quality and leverage great insights from the big data. In the next step, we will continue to investigate the deep learning classification algorithm by either adding more layers of convolutions into the current shallow CNN structure or using other deep learning architectures such as the LSTM and bi-directional LSTM networks for document representation. Also, we hope to be able to get more data from the hospitals to completely demonstrate the power of AI in big EHR analysis.

ACKNOWLEDGEMENTS

The authors would like to thank the support from both the Technology Research Centerand the Business Unit of Medical Health in the Software Business Group of Inspur Inc, and the Health Planning Commission of Shandong Province, China.

REFERENCES

[1] HanLP,https://datascience.shanghai.nyu.edu/hanlp.

[2] Mikolov Tomas, Chen Kai, Corrado Greg, Dean Jeffrey (2013) "Efficient Estimation of Word Representations in Vector Space", Advances in neural information processing systems, pp3111-3119.

[3] Le Quoc, Mikolov Tomas (2014)"Distributed Representations of Sentences and Documents", Proceedings of 31 International conference on machine learning, pp1188-1196.

[4] V. N. Vapnik (1999)"An overview of statistical learning theory", IEEE Trans. Neural Network, vol. 10, pp988-999.

[5] Yoon Kim (2014)“Convolutional Neural Networks for Sentence Classification”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp1746-1751.

[6] Alex Krizhevsky, Ilya Sutskever, Geoffery Hinton (2012) “ImageNet Classification with deep convolutional neural networks”,NIPS'1, Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, pp1097-1105.

[7] https://en.wikipedia.org/wiki/Tf–idf

AUTHORS

Junmei Zhong received the B.Sc. degree in Computer Science from Dalian University of Technology, China, in 1988, the Master’s degree in Computer Science, Nankai University, Tianjin, China, in 1993, where he received the “Excellent Thesis Award”,the Ph.D. degree from Electrical & Electronic Engineering, The University of Hong Kong in 2000, where he received the prize of "Certificate of Merits for Excellent Paper", Awarded by IEEE Hong Kong Section and Motorola Inc, Dec. 1998.

He has been the Chief Data Scientist at Inspur USA Inc since March 2017. His R&D interests include machine learning, data mining, NLP, text mining, digital advertising, graph theory, knowledge graph, deep learning, signal processing, wavelets, image analysis, pattern recognition, and computer vision.

Before joining Inspur USA Inc, He was the Senior Principal Data Scientist at Spectrum Platform Company and Twelvefold Media Inc for content-based display advertising, Principal Data Scientist at Pitchbook Data Inc about NLP and text mining. Dr. Zhong was the research faculty in University of Rochester, NY, USA from 2002 to 2004, and Assistant Professor in Cincinnati Children's Hospital Medical Center, Ohio, USA from 2004 to 2006. He has generated many scientific papers published on prestigious journals and top conference proceedings.

Xiu Yi received the B.Sc. degree from Dept. of Computer Science and Technology, Harbin Engineering University, Harbin, China, in 2009, the Master’s degree in Computer Application Technology, Harbin Engineering University, Harbin, China, in 2012.

Since 2014, she has been a software engineer in the Technology Research Center of Inspur Software Business Group Co. Ltd., Jinan, Shandong Province, China, for machine learning, data mining, NLP, deep learning, text mining, and computer vision in OCR, face recognition and auto license plate recognition. From 2012 to 2014, she was a software engineer in Baidu, Beijing for NLP, NER and relation prediction with knowledge graph. She is proficient in C, Java and Python programming.

Jian Wang received the B.Sc. degree in Applied Mathematics from Shandong University, China in 2014, the Master’s degree of Science from Shandong University,Jinan, Shandong Province, China in 2017.

Since 2017, he has been an assistant engineer in the Technology Research Center of Inspur Software Group Co. Ltd. His current work is mainly aboutNLP, data mining, machine learning,deep learning and software development in Python for big data analysis.

Zhuquan Shao received B.Sc. degree in Applied Mathematics from Dezhou University, Shandong Province, China in 2014, the Master's degree of Science from Dalian Maritime University, Dalian, China in 2017.

Since 2017, he has been an assistant engineer in the Technology Research Center of Inspur Software Business Group Co. Ltd. His current work is mainly in machine learning, big data analysis, data warehouse, and software development in SQL and Python.

Panpan Wang received the B.Sc. degree with the major of Statistics from QuFu Normal University, Shandong Province, China in 2014, the Master's degree of Science from Dalian University of Technology, Dalian, China in 2017.

Since 2017, She has been working in the Technology Research Center of Inspur Software Business Group Co. Ltd., Jinan, Shandong Province, China. Her current work is mainly about deep learning and machine learning in the field of medical healthcare.

Sean Lin received the B.Sc. degree in Environment and Design from University of Jinan, Jinan, Shandong Province, Chinain 2007.

Since 2007, he has been the principal engineer in user experience design, big data analysis, project management, visualization and data governance. He worked in Guiyuan Tech Ltd. in Jinan, China, for project management and UI design from 2007 to 2010. He received the Certificate of System Integration and Project Management Engineer in 2014.

INTENTIONAL BLANK

MAKING MDD AGILE THE AGILE MODEL-DRIVEN METHOD

Klaus Mairon[1], Martin Buchheit[1], Martin Knahl[1] and Shirley Atkinson[2]

[1]Faculty of Business Information Systems, Hochschule Furtwangen University, Furtwangen, Germany
[2]Centre for Security, Communications and Network Research, Plymouth University, Plymouth, United Kingdom

ABSTRACT

This article takes up the idea of model-driven development in a new way and analyses existing points of criticism of this approach, which is well established in practice. The advantages of model-driven development seem obvious on the one hand, on the other hand there is criticism of the practicable use and the accusation of missing suitable process models. This environment of professional software development is currently characterized by the use of agile process models such as Scrum, XP, etc. However, an agile process model for the use of model-driven development (MDD) does not yet exist. An analysis of the similarities between existing approaches to MDD process models and existing agile modelling techniques forms the basis for the definition of a new agile process model. The Agile Model-Driven Method (AMDM) is the result of these studies.

KEYWORDS

Model-Driven Development, Model-Driven Architecture, Process Model, Agile Method, Software Engineering, UML

1. INTRODUCTION

Model-driven development has been a well-established term in modern software engineering for years. With the Model-Driven Architecture (MDA) defined by the Object Management Group (OMG) back in 2001, a defined industry standard exists for this purpose. It defined a uniform basis for the underlying technologies and modeling languages and thus responded to the trend that model-driven software development (MDSD or MDD) and model-driven engineering (MDE) became increasingly relevant for professional software development [28][33][40]. The MDA provides for a multi-stage transformation from the Computation Independent Model (CIM), through the Platform-Independent Model (PIM) and Platform-Specific Model (PMS) to implementation. The CIM is the model with the highest abstraction level and is the starting point for the other model types, the PIM and PSM. The Platform-Independent Model describes the structures and functional requirements of an application and concretizes the CIM without taking technological aspects of the target platform into account. Only the Platform-Specific Model enriches this model with additional model elements (e.g. stereotypes, tagged values) and defines the final transformation into the implementation [18][32]. The MDA provides the Unified Modeling Language (UML) as the modeling language, whereby any modeling language based on the OMG standards MetaObject-Facility (MOF) and Common Warehouse MetaModel (CWM) is allowed [34]. The UML will be extended by so-called profiles, which enriches the modeling

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 105–118, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80508

language with the model elements necessary for the business or technical domain. The result is a domain-specific language (DSL) based on UML (or another modeling language).

While the MDA covers the spectrum of model-driven development from the description of technicality to design, construction, deployment, operation and maintenance and describes all development stages via models, this has not been accepted in the practice of software development.

It is true that the model with the functional requirements is also the focus here. But it is often a Platform-Independent Model based on a technically motivated DSL or a mixture of Platform-Independent and Platform-Specific Model, which already contains rudimentary information for the transformation to the target platform [45]. The actual platform-specific model is often not modeled, but the transformation into the code (model-to-artefact transformation) takes place directly on the basis of the PIM. The background to this is the simpler way of incorporating functional changes into the model during the development process, without having to adapt another model - the PSM - accordingly. The automated model-to-model transformations from PIM to PSM, which are otherwise necessary for this purpose, often do not work in round-trip or reengineering or are too complex. The transformation to the target platform, i.e. to the chosen architecture, programming language, test definitions and documents is implicitly carried out in one step of the model-to-artefact transformation (often by a corresponding generator framework). In practice, this is often quite sufficient, since the target platforms rarely change fundamentally and several software products are produced based on a defined form. Examples of this can be a collection of JEE components for a logistics solution or several web and/or microservices for insurance companies. This is also referred to as software product families, i.e. building blocks of different functional contents which, however, all correspond to the same design principles.

Although OMG has laid the foundations for the use of MDD in professional software development by creating the necessary standards, the technology is currently not widespread. Even the positive factors of model-driven development, which Hutchinson et al. [27] name, could not change this. This not only includes a more efficient realization, but also a higher quality and faultlessness of the produced product. Hutchinson et al. see a long history of bad experiences (e. g. with CASE) as the cause. The following chapter identifies and describes further negative influences on the spread of model-driven development.

The lack of suitable process models for MDD as well as the additional complexity and a high initial effort are identified as additional problems. For this reason, we finally present an agile, model-driven process model which is suitable to reduce these influences.

2. PROBLEMS IN THE CONTEXT OF THE MODEL-DRIVEN APPROACH

A more efficient and high-quality software development should be in the interest of the software industry. So what are the reasons for the restrained use of model-driven software development? The following points can be identified:

(a) In their review, the authors Asadi and Ramsin point out that model-driven software development makes no sense without being embedded in a corresponding and supporting process model [5]. And in the scientific environment, there are individual process models for model-driven development. Thus, there are process models, e.g. the ODAC methodology [20] [21], MASTER [31], C3 [26], DREAM [44], MODA-TEL [19] and DRIP Catalyst [22], which all support the model-driven software development. However, Asadi and Ramsin critically point out that the process models they examined did not adequately support software engineering activities and did not consider overlapping activities.

In addition to Asadi and Ramsin, Chitforoush et al. concluded in [10] that, in principle, there are very few methods for the use of model-driven software development and the description of the processes is usually very incomplete and imprecise.

(b) Another problem is the high initial effort that characterizes an MDD project. This starts with the preparation of the necessary infrastructure, and extends from the development of the DSL, the corresponding metamodels to the definition of the necessary transformations to the target platform. This is referred to as "MDD infrastructure" or "Domain Architecture" in the literature (cf. [45]) and in the process models described in (a). And it is also described as a necessary step in the development process, which is time consuming and costly.

(c) Hailpern and Tarr identify additional problems that arise in the context of model-driven development [23]. For example, the concept of the different viewpoints and views leads to redundancies in the models. This results in manually created, duplicate work and makes a consistency management necessary. In addition, the authors criticize the complex relationships between the different levels of abstraction. Any change to an artefact will have corresponding effects on one or more other artefacts on other levels. This leads to massive problems in round-trip engineering. The third point of criticism from Hailpern and Tarr is the additional complexity resulting from an increasing number of artefacts, their increasing number of relationships with each other and the necessary use of tools. This means that they see massive problems with future maintenance, troubleshooting or alteration of the created artefacts. Finally, they fear an additional problem in the diversity of modeling languages. The standardization of the UML should be an important basis for the success of MDD. However, with the powerful extension capabilities of the UML through the Meta Object Facility (MOF) [35], a variety of dialects have been created. This makes the semantically correct use of the UML difficult, also for tool vendors to provide supporting technologies.

Heijstek and Chaudron studied in [24] these factors as part of a large industrial software development project and confirm the factors mentioned. For example, a code generator is another application that needs to be developed, tested and maintained. This means initial effort and increased complexity in the tool use by dependencies. It is therefore not surprising that Singh and Sood in [43] also make the future use of MDD in industrial software development dependent on the fact that MDD must be fully integrated into a software development process. In addition, the tool support is a major success factor and the complexity of the modeling language with its additionally required knowledge is a potential barrier.

During this research, these assessments were confirmed by three case studies from practice. These were three projects from the business environment of insurance companies. Here it was also found that there is certainly potential for the promised increase in efficiency and quality. However, various problems in dealing with MDD could also be confirmed. In one case, the development of the "Domain Architecture" was discontinued due to the high effort and the project was developed traditionally. In another case, the consequences of the high complexity and dependencies between the artifacts are evident, which often leads to project delays. In addition, these experiences were also confirmed by case studies by IBM [8][11], ABB Robotics and Ericsson [48], Autoliv, Sectra and Saab Aerospace [17] and Motorola [6].

3. AGILITY AS A POSSIBLE SOLUTION

An iterative approach in the development of the "Domain Architecture" as well as a stronger interaction with the actual application development was sketched in a case study (see above) as a possible solution. As a result, parts of the "Domain Architecture" would be available for application development earlier. It would also contribute to the exchange of experience between application developers and MDD infrastructure providers. In addition, application development is enabled to achieve results for end customers at an early stage. These are properties that are already well known in agile software development and are well established in the industry. Methods such as eXtreme Programming [7], Scrum [15][41][42], Crystal [13][14], Adaptive Software Development [25], DSDM [46][47], or Feature Driven Development [3][12][37] have been established in the practice of software development [38] and have also been adapted to large projects by methods such as LeSS [30].

However, before we can think about using agile concepts in model-driven software development, we must first study the extent to which agile techniques for modeling exist. In practice, the agile principles are often misinterpreted by many developers as an order to code directly and not to document (or to model). Scott Ambler and Mark Lines, however, have defined the concept of agile modeling in [4] and described some agile modeling techniques such as “Model storming”, “Iteration modeling” or “Architecture envisioning” in [1].

There are also some agile process models that include modeling as a phase or work step, but do not fully implement the model-driven approach with its support of automated artefact creation. They often see the modeling "only" as a means of communicating with the customer and describing the functional requirements. These process models include:

(a) Agile Model Driven Development (AMDD): Scott Ambler describes an approach to integrate modeling in agile software development in [2]. However, Ambler places the focus on creating models with a minimum of effort and keeping them as simple as possible. He only wants to identify the most necessary requirements. This includes requirements for architecture as well as the basic functional requirements. However, this approach shows that AMDD is not a model-driven approach in terms of MDA or MDD and the use of their concepts. The use of tools for modeling and the automated transformation into various other artefacts is not applied. Instead, we must speak of a model-based approach.

(b) Feature-Driven Development (FDD): FDD was first described by Peter Coad et al. in [12] as a lean method for software development. The method provides the notion of "features" in the center of development. A "feature" is defined as a property of an application that is useful in the eyes of the customer. Unlike other agile process models in Feature-Driven Development modeling is a defined activity in the process model. So already in the first step of the process, an overall model is created. The aim of this first step in the process is to achieve a common understanding of the content and scope of the system under development. Another major step in the process flow, which is supported by modeling, is the design of a feature. During a walk-through, the chief programmer is developing along with the feature team a refined model. The design of the feature is checked during the inspection. But FDD is not a process for implementing MDD approaches. On the one hand, the definition of an architecture is not provided at all in the development process; on the other hand, manual coding by programmers is the focus.

(c) MIDAS Framework: In their comparison of different MDA-based methods, Chitforoush et al. [10] and Parviainen et al. [39] mention the MIDAS framework. MIDAS should support the agile development of web information systems. For this MIDAS uses UML as modelling language for the creation of the necessary PIMs and PSMs. In addition, MIDAS defines mapping rules for the transformation of models from PIM to PIM, PIM to PSM and PSM to PSM. However, unlike the other MDD methods MIDAS defines no concrete development process. In another paper, Caceres et al. describe the experiences they have had in a case study with the integration of agile practices and activities from XP in MIDAS [9]. According to the authors, it turned out to be positive, to develop the CIM (Computation Independent Model) as an early general vision of the future application. In addition, MIDAS uses various agile techniques such as pairwise development or continuous integration. Based on their case study, the authors conclude that it is important to identify the strengths of agile modelling, to guide developers in creating the models, and to make a breakdown of the different requirements.

Of the three presented process models, only MIDAS can actually establish a relationship with MDA and model-driven software development. However, here too there is criticism about a missing development process.

4. DEVELOPMENT OF AN AGILE APPROACH FOR MDD

As outlined in sections 2 and 3, there are various rudimentary approaches for a process model for model-driven development. MODA-TEL, MASTER, etc. are oriented more towards the classical approach, while MIDAS, for example, tries to implement agile concepts. In the following, an agile process model for model-driven software development is presented, which is based on these basic structures and fully reflects the aspects "Domain Architecture", "Development Process" and "Team / Roles". It is designed to support the agile development of small to medium business applications through model-driven development. To develop this process model, the common elements of the individual existing process models were first identified and abstracted, and then described using metamodels. The thus obtained metamodels allows the comparison of the individual methods and the identification of gaps. Metamodels were defined for the description of the process steps, the team roles and the artefacts arising in the process. Fig. 1 shows, for example, the identified artefacts in the model-driven development. Some examples of such artefacts include:

- the domain-specific language, which is defined by a metamodel and describes the elements of the problem domain; and
- a reference model that uses the defined DSL and
- the associated reference implementation, which enables the derivation of model-to-artefact (or model-to-code) transformations based on the application architecture.
- A generated prototype enables a verification of the transformation against the reference implementation.

There was an additional focus on the identification of the dependencies between the individual artefacts and the possibility to further develop and refine them iteratively and incrementally without taking too much influence on the dependent artefacts. The goal was to provide the "Domain Architecture" in its early stages for the development of the application. However, the "Domain Architecture" will be changed during the project by

- the identification of other non-functional requirements affecting the application architecture and architecture and

- the extension of the domain-specific language by additional language elements.

The affected artefacts must therefore depend exclusively on the resulting models and transformation rules and must never depend on manual extensions. In addition to the definition of a new process model (i.e. a further instantiation of the developed metamodel), it was also examined which agile working techniques appear to be suitable for this process model in dealing with models and how these can be integrated.

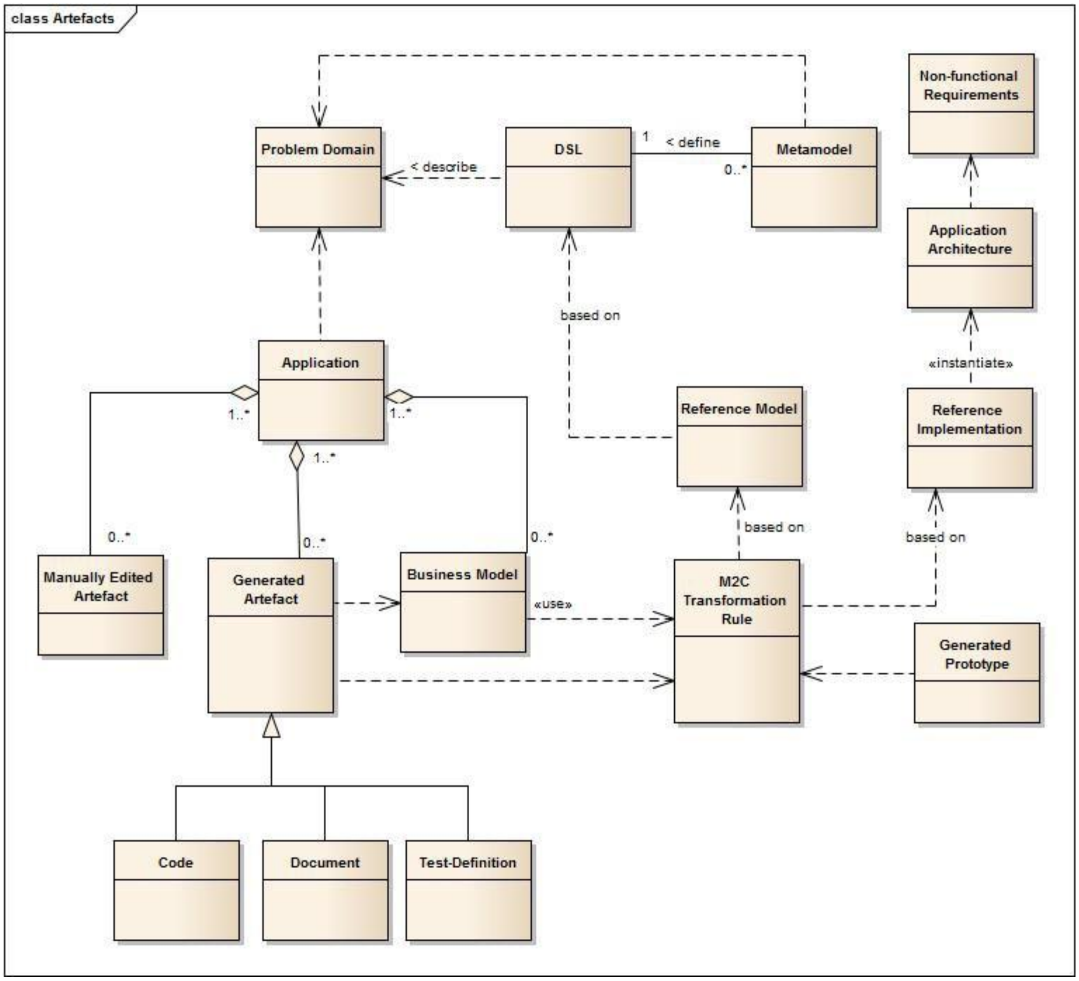


Figure 1: Artefacts in Model-Driven Projects.

5. THE AGILE MODEL-DRIVEN METHOD

As learned from the studies of Asadi and Ramsin [5] and Chitforoush et al. [10], the existing methodologies for MDD projects are incomplete and their description is imprecise. Essentially, they are based on traditional development processes, and the process framework by Chitforoush or the development lifecycle of Asadi and Ramsin do not regard agile aspects. Other approaches like AMDD [2] focus on the use of models in agile methods, but they do not consider MDD.

Based on the developed metamodels, the Agile-Model-Driven Method, "AMDM", was defined as a new process model for agile model-driven development. The individual aspects "Process", "Team" and "Domain Architecture" will be briefly outlined below. The terms often refer to elements of Scrum or XP, since these approaches are widely used in practice [29] and their underlying principles are established. Here, for example, the iterations are also referred to as sprints (cf. SCRUM [15]).

5.1 Process

The development process of AMDM is divided into three types of sprints as well as a parallel optimization of the software architecture (see Fig. 2).

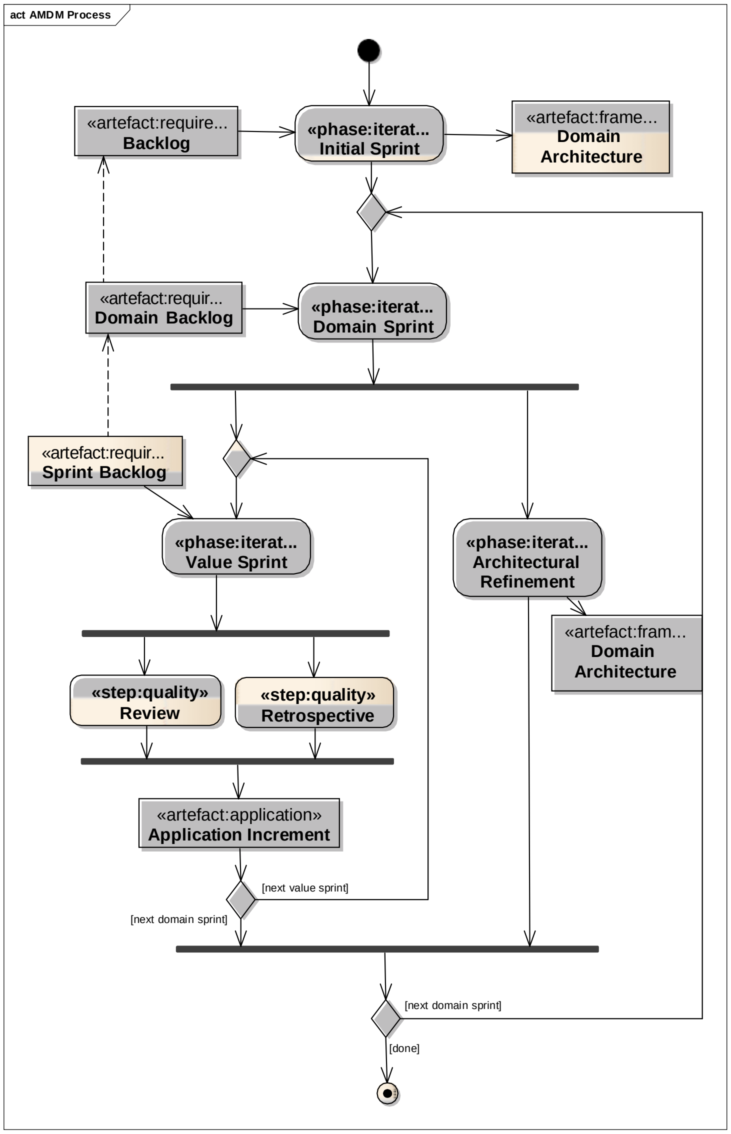


Figure 2: The Process in the Agile Model-Driven Method

The initial sprint defines the key requirements and the basics for the project. The backlog is created with the basic functional and non-functional requirements. Based on this, the rough application architecture is defined as a "big picture". In addition, a first version of the MDD infrastructure (Domain Architecture) with the DSL, the first transformation rules and the necessary frameworks as well as a minimal prototype (walking skeleton) are created here. The initial sprint is followed by a domain sprint where the DSL is defined in relation to a selected subset of the problem domain. Within this domain sprint, the DSL and transformation rules for a specific aspect of the application are defined.

The domain sprint is followed by one or more value sprints in which the requirements from the backlog of the problem domain are modelled with the DSL, generated and implemented using the corresponding transformation rules. Following a value sprint, a review of the results as well as a retrospective of the process are given. This serves to continuously increase the quality of the results. Parallel to the development in value sprints, a refinement of the application architecture takes place considering the individual non-functional requirements or the findings from the retrospectives. A domain sprint with an adaptation of the DSL or the transformation rules follows upon a sequence of value sprints with the implementation of the technical requirements.

5.2 Team

Teamwork and communication is a main aspect in all agile process models. In AMDM the team works interdisciplinary. Each team member has his own know-how and contributes to the project success. In AMDM, the team is divided into three main groups:

(a) The first group knows and understands the functional requirements of the business application. On the one hand, there is the typical product owner, who represents the customers view in the project and names the requirements and prioritizes them. In addition, however, there are those project staff who model the problems using the domain-specific language. The concrete roles are named Product Owner and Business Analyst.

(b) The second group defines the architecture of the application and the domain architecture for the model-driven development. They define the domain-specific language formally by developing a meta-model and create the necessary rules for the model-to-artefact transformations. The roles are named Application Architect, Domain Architect, Domain Developer.

(c) The third group is the group named Application Developers who, according to the architecture specifications, manually supplement the generated source code with non-generateable functionality. During the Initial Sprint or Domain Sprint they also develop an application prototype as a “walking skeleton”.

5.3 Domain Architecture

The domain architecture deals with the definition of the domain-specific language (DSL) based on a metamodel as well as the definition of the necessary rules for the model-to-artefact transformation. The metamodel describes the elements of the domain-specific language. Depending on the chosen modeling language (e.g. UML) and diagram form (e.g. class diagram, activity diagram), it relies on the associated metamodels. In the case of UML, this is MOF [32][35][36]. The business analyst and domain architect as well as the domain developer are involved in the development of the metamodel. The business analyst describes the terms of the problem domain; the domain architect develops the corresponding metamodel and explains the meaning of the model elements to the domain developers. For the required model-to-artefact transformation, the knowledge of the application architecture is also necessary, additionally to the understanding of the DSL. For this purpose, the Domain Architect and the Application Architect work on the definition of the transformations. Because the process enables the iterative and incremental development of the application architecture parallel to the development process, it ensures that not too many non-functional requirements need to be considered at the same time and that the complexity increases gradually.

6. APPROACHES TO EVALUATION AND VERIFICATION

In the previous paragraph the core aspects of the Agile-Model-Driven-Method were presented. However, it is difficult to verify the approach in a practicable way. One possibility to verify AMDM is the projection on the examined case studies. Initially, the following problem areas of model-driven development were mentioned:

(a) High initial effort: AMDM is an iterative and incremental development process. The domain architecture as the basis of the model-driven development is created successively. The waterfall-like approach to create all MDD artefacts at the beginning would have been avoided. In addition, AMDM is designed to produce results continuously. The development of a software application can be supported much earlier.

(b) Management mistrust: Confidence in the technology of model-driven development can be created by the early and continuous provision of usable results. Overall, the development of software becomes more efficient and of higher quality.

(c) High complexity and many dependencies: By focusing on the problem domain in the domain-specific language, AMDM avoids mixing technical with non-technical elements. The architecture is developed in parallel with the modeling and implementation of business requirements. Adjustments resulting from architectural changes do not affect the models, but only the transformation rules. And because of the evolutionary development of the architecture, their changes are always limited.

Additional suggestions from the other case studies of ABB Robotics and Ericsson, IBM, Motorola etc. were also considered and evaluated:

(a) AMDM involves business analysts, architects and developers in the development of the domain-specific language, the transformations and the implementation of the requirements. The work of the teams overlaps, so that the respective sub teams always have the necessary know-how.

(b) AMDM assumes that parts of the application must be encoded manually. This is done in tight cycles within a value sprint.

(c) The use of concepts of known and established agile methods reduces the inhibition threshold for the use of AMDM. The scepticism towards model-driven development can be counteracted by the early availability of partial results.

(d) AMDM also causes additional costs by defining domain-specific language and transformation rules. However, the early provision of applicable partial results at an early stage will bring an early benefit. The quality of automated application development reduces future error analysis costs. In addition, the architecture is being further developed in an evolutionary and iterative manner. And only as far as it's necessary. This also eliminates unnecessary costs for the creation of a bloated architecture and the resulting complexity of the model-to-code transformation.

(e) The case study by Elmqvist and Nadjim-Tehrani [17] confirms that model-driven development leads to cost savings in manual code implementation. However, they are concerned about the availability of adequate tools for the entire development process, from specification to implementation. In the meantime, however, there are sufficient tools available, from UML modeling tools to generators, which are based on the relevant OMG standards. In AMDM, these tools can be used in an agile process. From the description of the requirements to the evolutionary development of the architecture and the definition of the domain-specific language.

(f) IBM's case studies [8][11] also confirm the potential of model-driven development. However, they assume that many manual changes to the models and the generated source code remain necessary. AMDM tries to keep the business models as stable as possible by basing them on the domain-specific language. However, IBM's studies also assume a pure MDA approach and a two-stage transformation from PIM to PSM to source code. AMDM does not follow this approach. AMDM uses a direct transformation from the commented PIM (based on DSL) directly into the source code.

(g) Likewise, criticisms from the Motorola study [6] are considered. In AMDM the intensive communication between architects, domain architects, business analysts and developers ensures that the knowledge about the models, transformations and the target architecture are evenly distributed in the team. This avoids implicit or explicit assumptions about implementations.

Finally, we have to assess whether AMDM is agile. It can be said:

(a) Customer Involvement: In AMDM, the customer's interests are represented by the product owner analogously to Scrum. He takes up new requirements, prioritizes them and leads them to the development process.

(b) Incremental delivery: This aspect has already been discussed before. Frequent partial deliveries are supported.

(c) People not process: AMDM also focuses on the communication and efficient collaboration of team members. However, due to the additional complexity of the model-driven development, the process is more strongly emphasized than in other agile approaches.

(d) Embrace change: Openness towards changes is also implemented in AMDM. Functional and non-functional changes can be included in the development at any time.

(e) Maintain simplicity: KISS (keep it simple, stupid) is a basic principle in the evolutionary development of software architecture, because the degree of complexity is growing in proportion to the requirements.

Looking at the sum of these criteria and comparing them to the Agile Model-Driven Method, this can be justly described as agile.

7. CONCLUSION AND FURTHER RESEARCH

The Agile Model-Driven Method combines agile working techniques with the development approach of model-driven development. For this, the elements of model-driven development were first identified and the existing limits and risks were considered. The criticism and scepticism of the model-driven development, which has often been expressed in practice, has also been analysed. Case studies from specific projects in the field of the author as well as case studies from other branches were used for this purpose. Thus, potentials and criticisms of the model-driven development could be identified.

For the definition of an agile model-driven development methodology it was necessary to characterize the project phases of an MDD project, the involved roles and artefacts. For this purpose, common features of existing process models for model-driven development were identified and described based on a metamodel. On this basis and considering appropriate agile modeling techniques, the Agile Model-Driven Method has been defined. It enables an agile model-driven development of business applications in a continuous process, from the specification of the requirements to the implementation. It fulfills the criteria of an agile approach and is designed to minimize the problems and criticisms encountered in the investigated case studies.

AMDM is particularly suitable for the development of small to medium-sized business applications based on standardized components or services. The problem domain of these applications is limited and well structured so that the requirements and domain-specific language can be broken down into features. In addition, the experience with agile software development is widespread in this environment. The same applies to the principles of model-driven development based on MDA or MDD, even if they are rarely used. The limits of AMDM are reached when developing specific applications with very individual components. Here, manual implementation and optimization is still the method of choice. A possible additional difficulty is probably scaling to larger or distributed teams. The role of a mediator is recommended for this purpose, as described by Jutta Eckstein in [16].

Finally, the AMDM method must be applied in practice. Only through the use in small and medium-sized projects comparable to the showcases described above, further questions arise which can be examined.

Another open point, which can only be answered by appropriate experience, is the effort assessment and thus the planning of the sprints: How does the combination of modeling, generation and manual coding affect the effort required to implement a user story? Is the assumed timeframe of the typical two weeks for a sprint sufficient or even too long in this case?

The quality assurance of models can be a further focus of research. How can they be validated and verified? This is an independent field of research, but its results may be interesting for AMDM to conduct reviews.

And finally, another topic may be the classification of domain-specific languages. These are increasingly defined as text-based domain-specific modeling languages. Here an investigation of the different types of DSLs with an assessment of the suitability for different problems would be helpful.

REFERENCES

[1] Ambler, S., Jeffries, R. (2002). *Agile modeling. Effective practices for eXtreme programming and the Unified Process.* New York, NY: Wiley.

[2] Ambler, S., (2004). *THE OBJECT PRIMER. Agile model-driven development with UML 2.0*. New York: Cambridge University Press, 3rd Edition.

[3] Ambler, S. (2005). Feature Driven Development (FDD) and Agile Modeling. [online] http://www.agilemodeling.com/essays/fdd.htm [08/10/2017].

[4] Ambler, S., Lines, M. (2012). *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, Boston: Pearson Education.

[5] Asadi, M.; Ramsin, R. (2008). MDA-Based Methodologies: An Analytical Survey. In: Ina Schieferdecker, Alan Hartman (Eds.): *Model Driven Architecture – Foundations and Applications:* Springer Berlin / Heidelberg (Lecture Notes in Computer Science, vol 5095), pp. 419–431.

[6] Baker, P., Loh, S., Weil, F. (2005). Model-Driven Engineering in a Large Industrial Context - Motorola Case Study. In: *Model Driven Engineering Languages and Systems, 8th International Conference, MoDELS 2005*: Springer, Berlin / Heidelberg (Lecture Notes in Computer Science, vol. 3713), pp. 476 – 491.

[7] Beck, K. (2003). *Extreme programming explained. Embrace change*. Boston: Addison-Wesley, 8th print.

[8] Brown, A., Conallen, J., Tropeano, D. (2005). Practical Insights into Model-Driven Architecture: Lessons from the Design and Use of an MDA Toolkit. In: Beydeda, S., Book, M., Gruhn, V. (Eds.): *Model-Driven Software Development.* Berlin, Heidelberg: Springer, 1st ed., pp. 403–431.

[9] Cáceres, P., Diaz, F., Marcos, E. (2004). Integrating an Agile Process in a Model Driven Architecture. In: GI *Jahrestagung 2004*, pp. 265–270.

[10] Chitforoush, F., Yazdandoost, M., Ramsin, R. (2007). Methodology Support for the Model Driven Architecture. In: *Software Engineering Conference, 2007.* APSEC 2007. 14th Asia-Pacific, pp. 454–461.

[11] Chowdhary, P. et al. (2006). Model Driven Development for Business Performance Management. In: IBM *Systems Journal* (Vol. 45, No 3), pp. 587–605.

[12] Coad, P., Lefebvre, E., Luca, E. de (1999). *Java modeling in color with UML. Enterprise components and process*. Upper Saddle River, NJ: Prentice Hall PTR.

[13] Cockburn, A. (2001). *Agile Software Development*. Reading, Massachusetts: Addison-Wesley.

[14] Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams.* Boston: Addison-Wesley.

[15] Cohn, M. (2010). *Succeeding with Agile. Software development using Scrum*. Upper Saddle River, NJ: Addison-Wesley.

[16] Eckstein, J. (2010). *Agile software development with distributed teams. Staying agile in a global world.* New York: Dorset House Pub.

[17] Elmqvist, J., Nadim-Tehrani, S. (2005). Intents and Upgrades in Component-Based High-Assurance Systems. In: Beydeda, S., Book, M., Gruhn, V. (Eds.): *Model-Driven Software Development*. Berlin, Heidelberg: Springer, 1st ed., pp. 289–303.

[18] Frankel, D. (2003). *Model driven architecture. Applying MDA to enterprise computing*. Indianapolis: Wiley (OMG Press).

[19] Gavras, A., Belaunde, M., Pires, L., Almeida, J. (2004). Towards an MDA-Based Development Methodology. In: Oquendo, F., Warboys, B., Morrison, R. (Eds.): *Software Architecture*: Springer Berlin / Heidelberg (Lecture Notes in Computer Science, vol. 3047), pp. 230–240.

[20] Gervais, M.-P. (2002). Towards an MDA-Oriented Methodology. In: *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment*. Washington, DC, USA: IEEE Computer Society (COMPSAC '02), pp. 265-270.

[21] Gervais, M.-P. (2003). ODAC: An Agent-Oriented Methodology Based on ODP. In: *Autonomous Agents and Multi-Agent Systems* 7, pp. 199–228.

[22] Guelfi, N., Razavi, R., Romanovsky, A., Vandenbergh, S. (2004). DRIP Catalyst: An MDE/MDA Method for Fault-tolerant Distributed Software Families Development. In: *OOPSLA and GPCE Workshop on Best Practices for Model Driven Software Development.*

[23] Hailpern, B., Tarr, P. (2006). Model-driven development: the good, the bad, and the ugly. In: *IBM Systems Journal* 45, pp. 451-461.

[24] Heijstek, W., Chaudron, M. (2010). The Impact of Model Driven Development on the Software Architecture Process. In: *Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on*, pp. 333–341.

[25] Highsmith, J. (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York: Dorset House.

[26] Hildebrand, T., Korthaus, A. (2004). A Model-Driven Approach to Business Software Engineering. In: *Proceedings of the 8th World Multi-Conference on Systemics. Cybernetics and Informatics*. Orlando, USA, pp. 74–79.

[27] Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S. (2011). Empirical assessment of MDE in Industry, ICSE 11, May 21-28, 2011, Waikiki, Honolulu, HI, USA. ACM, 2011.

[28] Kent, S. (2002). Model Driven Engineering. In: *Proceedings of the 3rd International Conference on Integrated Formal Methods*. London, UK, UK: Springer (IFM '02), pp. 286-298.

[29] Komus, A. (2014). *Status Quo Agile 2014. Study on success and forms of usage of agile methods.* University of Applied Sciences Koblenz. [online] http://www.hs-koblenz.de/en/rmc/fachbereiche/wirtschaft/forschung-projekte-weiterbildung/forschungsprojekte/status-quo-agile-en/ [10/09/2016]

[30] Larman, C. (2016). *Large-Scale Scrum: More with LeSS*. Boston: Addison-Wesley.

[31] Larrucea, X., Diez, A.G.; Mansell, J. (2004). Practical Model Driven Development Process. In: Akehurst, D.H. (Ed.): *Second European Workshop on Model Driven Architecture (MDA) with an emphasis on Methodologies and Transformations*. Canterbury, UK, 7th-8th September 2004. University of Kent, pp. 99–108.

[32] Miller, J., Mukerji, J. (Ed.) (2014). *MDA Guide Version 2.0*. Object Management Group (OMG). [online] http://www.omg.org/cgi-bin/doc?ormsc/14-06-01 [10/08/2017].

[33] Mohagheghi, P., Dehlen, V. (2008). Where Is the Proof? - A Review of Experiences from Applying MDE in Industry. In: Schieferdecker, I., Hartman, A. (Eds.): *Model Driven Architecture – Foundations and Applications*: Springer Berlin / Heidelberg (Lecture Notes in Computer Science, vol. 5095), pp. 432–443.

[34] Object Management Group (OMG) (2010). *The MDA Foundation Model*. [online] http://www.omg.org/cgi-bin/doc?ormsc/10-09-06 [10/08/2017].

[35] Object Management Group (OMG) (2011). *Meta-Object Facility (MOF) Specification, Version 2.4.1 (August 2011).* [online] http://www.omg.org/spec/MOF/2.4.1 [20/10/2011].

[36] Object Management Group (OMG) (2001). *UML Profile for Enterprise Distributed Object Computing*. Document ptc /2001-12-04.

[37] Palmer, S.R., Felsing, J.M. (2002). *A Pracitcal Guide to Feature-Driven Development*. Englewood Cliffs, NJ: Prentice Hall.

[38] Parsons, D., Ryu, H., Lal, R. (2007). The Impact of Methods and Techniques on Outcomes from Agile Software Development Projects. In: McMaster, T., Wastell, D., Ferneley, E., DeGross, J. (Eds.): *Organizational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda, vol. 235*: Springer Boston (IFIP International Federation for Information Processing), pp. 235–249.

[39] Parviainen, P., Takalo, J., Teppola, S., Tihinen, M. (2009). *Model-Driven Development. Processes and practices.* [online] http://www.vtt.fi/inf/pdf/workingpapers/2009/ W114.pdf. [08/12/2016]

[40] Schmidt, D.C. (2006). Model-Driven Engineering. In: *Computer 39 (2)*, pp. 25–31.

[41] Schwaber, K. (2004). *Agile Project Management with Scrum*. Seattle: Microsoft Press.

[42] Schwaber, K., Beedle, M. (2002). *Agile Software Development with Scrum*. Englewood Cliffs, NJ: Prentice Hall.

[43] Singh, Y., Sood, M. (2009). Model Driven Architecture: A Perspective. In: *International Advance Computing Conference (IACC 2009).* Patiala, India, 6-7 March 2009. IEEE.

[44] Soo Dong Kim; Hyun Gi Min; Jin Sun Her; Soo Ho Chang (2005). DREAM: A Practical Product Line Engineering Using Model Driven Architecture. In: *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 1, pp. 70–75.

[45] Stahl, T., Völter, M., Bettin, J., Czarnecki, K., Stockfleth, B. von (2006). *Model-Driven Software Development. Technology, Engineering, Management.* Chichester: Wiley.

[46] Stapleton, J. (1997). *DSDM Dynamic Systems Development Method*. Harlow, UK: Pearson Education.

[47] Stapleton, J. (2003). *DSDM: Business Focused Development*. Harlow, UK: Pearson Education. 2nd ed.

[48] Staron, M. (2006). Adopting Model Driven Software Development in Industry - A Case Study at Two Companies. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (Eds.): *Model Driven Engineering Languages and Systems, 9th International Conference, MoDELS 2006,* Genova, Italy, October 1-6, 2006, Proceedings: Springer (Lecture Notes in Computer Science, vol. 4199), pp. 57–72.

AUTHORS

Klaus Mairon is a part-time PhD student at the University of Plymouth and Furtwangen University (HFU). In his profession, Klaus Mairon is an IT consultant and software architect at msg systems ag as well as a lecturer at the Baden-Wuerttemberg Cooperative State University.

The co-authors Dr. Shirley Atkinson from the University of Plymouth, Prof. Dr. Martin Buchheit and Prof. Dr. Martin Knahl from Furtwangen University (HFU) supervise his doctoral thesis.

SOCCER EVENT DETECTION

Abdullah Khan[1,2], Beatrice Lazzerini[2], Gaetano Calabrese[3] and Luciano Serafini[3]

[1]Department of Information Engineering, University of Pisa, Pisa, Italy
[2]Department of Information Engineering, University of Florence, Florence, Italy
[3]Fondazione Bruno Kessler, Trento, Italy

ABSTRACT

The research community is interested in developing automatic systems for the detection of events in video. This is particularly important in the field of sports data analytics. This paper presents an approach for identifying major complex events in soccer videos, starting from object detection and spatial relations between objects. The proposed framework, firstly, detects objects from each single video frame providing a set of candidate objects with associated confidence scores. The event detection system, then, detects events by means of rules which are based on temporal and logical combinations of the detected objects and their relative distances. The effectiveness of the framework is preliminary demonstrated over different events like "Ball possession" and "Kicking the ball".

KEYWORDS

Event detection in video, simple events, complex events.

1. INTRODUCTION

Identifying intermediate and high-level complex events from an unstructured video is an extremely challenging task due to the variation and the dynamics of the video sequence. In this work, the focus is on the analysis of videos showing team sport activities and, more specifically, soccer game. Given the nature of the game itself, where two teams each of eleven players produce a vast number of possible interactions, soccer is a highly complex system [16]. Due to the high complexity governing the "beautiful game", the statistical analysis of soccer games has fascinated scientists and experts.

Data are playing an increasingly key role in sports, but they must be processed to extract meaningful information [2, 3]. Data-driven decision plays a significant role in soccer and many other sports. Collecting and properly handling quality data from a soccer match is, therefore, clearly of immense value for a team, management and other stakeholders.

The data typically collected from a soccer game include: goals scored, assists, number of shots on goal, possession information, corners, off sides, fouls, cards given, injuries, substitutions, etc. There is scope for the collection of larger data sets, such as the position-per-time of the ball, and each player on the field throughout the game, or on a short video clip. From this complex data set,

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 119–129, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80509

the objective is to detect specific and semantically meaningful events like player ball possession, team ball possession, kick or shoot, etc. Researchers from all over the world have been working for more than a decade to find different solutions for the video analysis. Their research in the domain of event processing is more focused on structured data. However, there are several applications for event driven systems based on image data. Therefore, there is a need for a system that can process multimedia events [1] from images and videos.

In this paper, the proposed framework attempts to detect different events. Images are given as an input to the object detector \Single Shot Multi-Box Detector" (SSD), which provides us with objects expressed in terms of bounding boxes with a given confidence score. We will use this system as a filter because the objects associated with confidence score higher than a specific threshold will be the input to the event detection system for detecting events. Then based on the distance between the bounding boxes of objects and using logical and temporal operators, events are defined.

2. RELATED WORK

Until the discovery of deep learning, sports video analysis, especially soccer video analysis, has been classified into two categories: object tracking and pattern recognition [21, 9]. The use of customized cameras [14] results in computational cost in case of object tracking, whereas the pattern recognition methodology simply extracts lower-level features and then uses a classifier to detect higher level events. A few methodologies which have been used with noticeable success for soccer activity recognition include: Qian et al [17] categorization of events into distinct categories like shoot, goal, etc. Such an approach includes feature extraction and heuristic rules for detecting events. They perform low-level analysis to detect marks (field, lines, logo, arcs, and goalmouth), player positions, ball position, etc, and then derive mid-level features using these cues. In the end, they developed a rule-based system to detect salient events like the goal, corner, etc. Jin et al [10] applied a Hidden Markov-based algorithm for video event detection based on cues fusion and integration. Detecting higher-level events from lower-level events is an important and challenging problem for soccer video analysis. The detection reveals, e.g., the movement of the players and the ball on the field, which could be used to identify certain actions ('passing the ball', 'shot on goal', etc.) or to better understand the overall trend of the game.

Since 2012, deep learning methods such as Convolutional Neural Networks and Restricted Boltzmann Machines have been successfully used for event and activity recognition. CNNs have shown better performance in image classification, object detection and modeling high-level visual semantics [11],[8],[6]; Recurrent Neural Networks have shown good results in modeling temporal dynamics in videos [12]. Frequently used action localization techniques, such as fast r-CNNs and faster r-CNNs [18],[7], usually start with the region of interest (proposal generation) to obtain a set of candidate regions, then use a fully connected layer at the end to classify objects.

Current approaches mentioned above focus on event recognition in soccer videos from the perspective of feature extraction, models, and classifiers for extracting low-level events. Such approaches lack the semantically meaningful representation of intermediate events. Injecting semantic definition and structural knowledge in these approaches is rather difficult. So, this motivates us to start from the basic building blocks and rebuild a system that allows exploiting the semantic knowledge about events, which can be used to recognize the intermediate and high-level complex events. To the best of our knowledge, while there are systems that automatically detect basic facts, like the position and the movement of the player, there are no automatic detectors for semantically complex events, like scoring on a penalty kick, or scoring on a corner kick.

The rest of the paper is organized as follows. Section 3 describes the video events as simple and complex events. Section 4 elaborates distinct types of events for the soccer scenario. In Section 5 the proposed architecture is highlighted and in Section 6 results and future work are presented, respectively. In section 7 we draw some conclusions.

3. VIDEO EVENTS

A precise ontological definition of event is still an open point. To the purpose of this paper we take the approach recently proposed in [4]. The main objective of this section is to precisely define the event structure we will adopt in our approach.

Video events can be defined as interesting events which capture the user attention [20] . For example, a soccer "shot on goal" event is defined as the ball kicked by a player and the ball moving towards the direction of the goal.

3.1 Simple Events

A simple event type is defined as follows:

$$SE = \langle ID, seType, t, \langle role_1, oType_1\rangle, \ldots, \langle role_n, oType_n\rangle\rangle \tag{1}$$

where *ID* is the identifier, *se*T*ype* is the event type, e.g. "*throwing the ball*", and *t* is the time instant in which the event occurs, $role_1$..., $role_n$ (n = 1, ,n_{max}) are the roles that different objects play in an event of this type, e.g. one role of simple event "*throwing the ball*" is the subject who throws and a second role is the thrown object; finally o*Type*$_i$ is the legal type of object that can play the role $role_i$, e.g., it is only players who can throw, and only balls can be thrown. Summing up, the complete definition of the event type "*throwing the ball*" is

$$\langle ID, Throwing\_the\_ball, t, \langle throwing\_Player, player\rangle, \langle throwed\_Object, ball\rangle\rangle$$

A specific instance of an event of simple type defined in (1) is the following tuple:

$$\langle ID, seType, t, \langle role_1, O_1\rangle, \ldots, \langle role_n, O_n\rangle\rangle$$

where *ID* is the event identifier, O_1 and O_n are identifiers of objects detected in the frame associated to the time *t*, respectively. The instance of "Throwing_the_ball"

$$\langle 12, Throwing\_the\_ball, t, \langle throwing\_Player, obj02\rangle, \langle throwed\_Object, obj01\rangle\rangle$$

describes a simple event of type "throwing the ball" that happened at time t, where the obj02 throws the obj01. Furthermore obj01 and obj02 are two objects detected in the frame corresponding to time t, of type ball and player respectively.

3.2 Complex Events

Complex events are built by appropriately aggregating events, previously defined. More precisely, starting from simple events, we can apply logical operators or temporal operators to build higher-level complex events. We can thus define the hierarchy of events, from the lowest level including the simple events to the higher and higher levels corresponding to more and more complex events. In the following, we define the two categories of complex events: logical complex events and temporal complex events.

- **Logical Complex events** A logical complex event stems from the application of logical operators like AND, OR, NOT to a set of events which may be simple or complex.

  $$LCE = \langle ID, ceType, t, L =< e_1 \; op \; e_2 \; op....op \; e_n > \rangle$$

  where *ID* is the event identity, *ceType* is the complex event type (such as "The goal is valid only if there is no foul"), *t* is the time instance in which the complex event occurs, *L* is the set of lower-level simple or complex events e_1. . . . . . e_n joined by logical operators *op* (*i.e. AND, OR, NOT*).

- **Temporal Complex events** A temporal complex event derives from the application of temporal operation THEN as follows:

  $$TCE = \langle ID, ceType, t, L =< e_1 \; THEN \; e_2 ...THEN \; e_n > \rangle$$

  where *ID* is the event identifier, *ceType* is the complex event type (such as “player 1 passes the ball to player 2"), *t* is the event occurrence time, *L* is the sequence of lower-level simple or complex events, e_1 ....e_n that must occur in the order. For example, e_1, e_2, e_3, e_4 may be, respectively, “player1 possesses the ball", “player1 kicks the ball", “the ball approaches player 2", “player2 gets in possession of the ball".

4. TYPES OF EVENTS

One of the most interesting things about soccer analysis is the ability to recognize events, such as a kick, goal, pass, offside, cards, ball possession, etc. from a common video. Most of the videos previously used in the event recognition use multiple fixed cameras to observe the position of all the players and the ball on the soccer field [5]. The use of such cameras improves the overall accuracy of the system for object tracking but they are computationally expensive. The fragment of video we have used can be easily accessible from the internet.

In this section, we try to define a few of the significant low or intermediate complex events in soccer video (consisting of a sequence of frames), such as ball possession and kicking the ball based on the distance between the bounding boxes of involved objects, and rules (combination of temporal and logical operators) defined for each event category.

In this first attempt we propose a rule-based definition of video events, but we are aware that this will turn out to be not very flexible, and in the future we will investigate on the possibility of automatically learning event detectors by using supervised machine (deep) learning techniques.

4.1 Ball possession Event

Ball possession can be classified as Player Ball Possession (PBP) and Team Ball Possession (TBP). Both have the same starting point but different end-points [13]. In our approach, only those time intervals in which the ball is in play are considered for determining the ball possession. When the ball is in play one of the two teams always has the ball possession. PBP starts immediately as soon as a player begins to perform an action with the ball and ends when the player is no more able to perform any action with the ball or there is game interruption.

Player ball possession can be formally defined as follows: the event occurs when the distance between a player and the ball is below a threshold value and that player is the nearest to the ball.

$$\begin{array}{l}
\left\langle ID, PlayerBallPossession, t+\tilde{k}, \langle PossPlayer, p_i\rangle, \langle PossObject, b\rangle \right\rangle \leftarrow \\
player(p_i), ball(b), D(p_i,b,t) < T_h, \\
\forall j \neq i, player(p_j), D(p_j,b,t) > D(p_i,b,t) \wedge \\
\forall k = 1 \ldots \tilde{k}, D(p_i,b,t+k) \approx 0
\end{array}$$

The event "Player Ball Possession" occurs at time $t+\tilde{k}$, when the distance *D* (p_i,b,t) between the player p_i and the ball *b* at time *t* is less than the threshold T_h , and the distance *D* (p_j,b,t) between the ball and any other player p_j, $j \neq i$, is greater than *D* (p_i,b,t). Also, after interaction, the distance between the player and the ball is very low for an appropriate number of $\tilde{k}$ consecutive frames. The value T_h determines the threshold value for a player being able to physically interact with the ball and must be calculated experimentally.

4.2 Kicking the ball Event

In the soccer video, with reference to the consecutive sequence of frames, the event corresponding to kicking the ball is identified, initially if the distance between a player and the ball is very low for a few frames. Then, if the distance between a player and the ball increases in an appropriate number of the subsequent frames and the player is no longer able to interact with the ball. We can formally define the event Kicking the ball as follows:

$$\begin{array}{l}
\left\langle ID, KickingTheBall, t+\tilde{k}, \langle KickingPlayer, p_i\rangle, \langle KickedObject, b\rangle \right\rangle \leftarrow \\
player(p_i), ball(b), D(p_i,b,t) < T_h \quad \wedge \\
\forall k = 0 \ldots \tilde{k}-1, D(p_i,b,t+k) < D(p_i,b,t+k+1)
\end{array}$$

The expression above holds true as long as the distance between the player and the ball increases after their interaction. T_h is the interaction threshold between the player and the ball. In a game Kick can be classified into several types: Free kick, Goal kick, Penalty kick, Corner Kick etc.

4.3 Limitations

While defining the events we are not considering all special cases that might occur during a match. In some cases, the player does not interact with the ball, and runs besides the ball without touching it. Player ball possession only starts with the first touch. Also, considering ball possession for the player nearest to the ball is wrong, e.g, when that player is standing with back to the ball. To better differentiate between kick or shoot and dribble, one can think of the speed with which the ball travels after the player ball interaction. For example, the speed of the ball after dribbling will be slower than that of kicking or shooting. We are also considering the same threshold for all the players as taking into account player profiles related to their typical interaction with the ball is out of the scope of this work.

5. PROPOSED ARCHITECTURE

Figure 1 describes the workflow for our methodology. The data at our disposal consist of approximately 5 mins long video, consisting of 7.5k annotated frames. Objects are detected from every single frame using SSD [15]. Then a specific threshold regarding the confidence score is defined to filter out the objects which are not required to define events. Finally, events will be detected based on the distance between the bounding boxes of objects using temporal and logical operators.

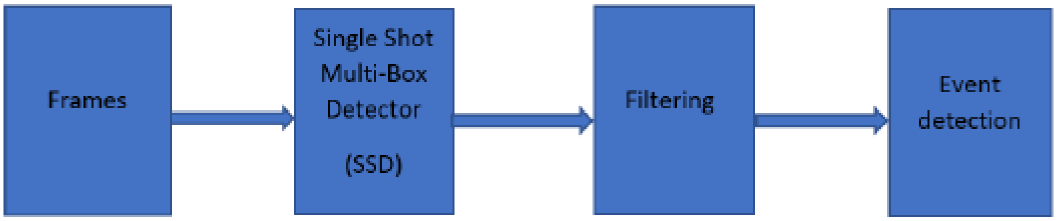


Fig.1. Block diagram of the proposed architecture

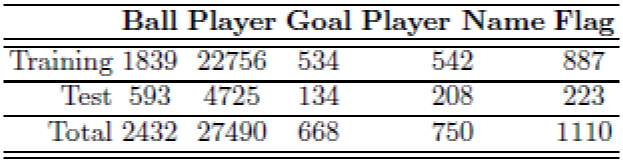
Frame Data We have a sequence of frames { f_1, f_2 f_n } Each frame is a set of bounding boxes, each bounding box gives us the position and dimension of an object, such as the ball or a player, by specifying the coordinates of the region containing the object. Frames are given as input to the SSD to detect objects with a confidence score.

Single Shot Multi-Box Detector (SSD) Most of the methods previously used for object detection have one thing in common, they have one part of their system dedicated to providing region proposals which includes re-sampling of pixels and features for each bounding box, followed by a classifier to classify those proposals. These methods are useful but are computationally expensive resulting in low frame rate. Another simpler way of doing object detection is by using a high-speed SSD system, which combines the two tasks of region proposal and classification in one system. The key idea behind SSD is small convolutional filters are applied to feature maps of bounding boxes to predict the category scores, using separate predictors for different aspect ratios to perform detection on multiple scales.

SSD needs an input image and ground truth for each object class during training. We have created this training set starting from a fragment of a real soccer match video, using Vatic [22], a Video Annotation Tool. Vatic allows annotating objects inside each frame drawing a bounding box around them. The output of this process is a set of images with relative bounding boxes coordinates saved in PascalVOC format.

Table 1 shows the numbers of object manually annotated, used for the training and test of SSD.

Table 1. Objects manually annotated to train and test the SSD



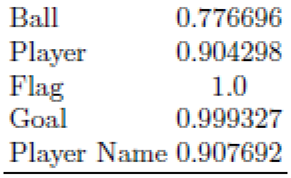
	Ball	Player	Goal	Player Name	Flag
Training	1839	22756	534	542	887
Test	593	4725	134	208	223
Total	2432	27490	668	750	1110

The training set in Table 1 has been used to create the SSD model. The average precision on the test set is given below in Table 2.

An example of the input image and the output image from the soccer match to SSD is shown in Figure 2 and Figure 3, respectively.

Filtering Filtering is performed by defining a specific threshold for the objects detected by the SSD. For example, as multiple players are detected in a single frame, then using a specific threshold, we can discard players in the frames which are not necessary to define the action.

Table 2. Average precision of the system



Ball	0.776696
Player	0.904298
Flag	1.0
Goal	0.999327
Player Name	0.907692

Event Detection System In many application domains, such as video event activity detection, sequences of events occurring over time need to be studied to summarize the key events from the video clips [19]. This section deals with the specific strategies adopted by the system for event detection. The steps involved are the detection and collection of the simple and low-level complex events, and the composition of the same to detect higher-level complex events. The system also includes an event type to identify the class of events. The new incoming event is registered within the system with a unique event identifier. The event recognition is performed by means of monitoring routines at two levels, low-level recognition and high-level recognition. The low-level event recognition involves detection of simple primitive events, while high-level event recognition handles detection of complex events. An event detection system receives, as an input, bounding boxes associated with a confidence score. Each bounding box also represents the coordinates of the object. To recognize the higher-level complex event, the system first detects simple and low-level complex events based on the rules defined for each event category and stores those events in the memory. We then apply logical and temporal operators on the detected events to recognize the higher-level complex events. Although there are several programming languages available to implement the event detection system, python was our preferred choice because of its highly intuitive general-purpose syntax.

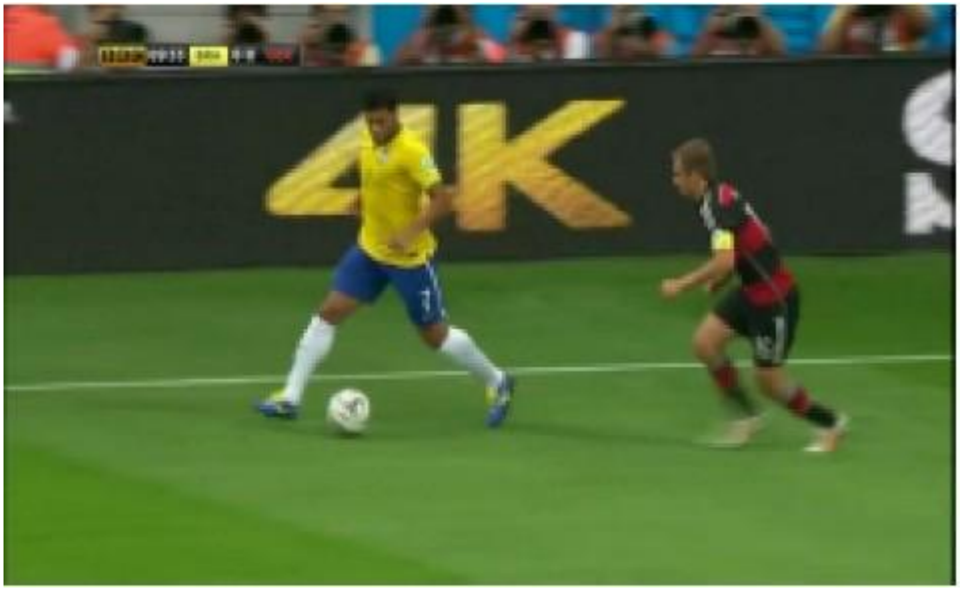


Fig. 2. Original frame

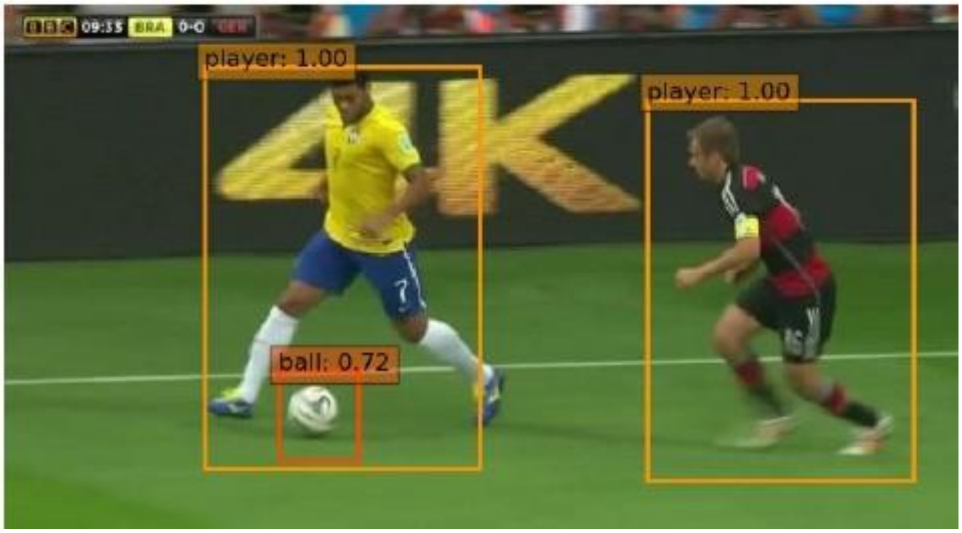


Fig. 3. Objects detected by SSD with confidence score

6. RESULTS AND FUTURE WORK

We have applied the proposed system to detect low-level complex events like "ball possession" and "kicking the ball" in the real soccer video. We have experimented on 5 minutes short video consisting of approximately 7.5k frames. We are aware of the fact that a limited number of events can be detected from this small data set. In the future, we will experiment on a larger data set, thus the number of events can be increased. Table 3 shows the event detection results. For Ball possession event, 13 out of 14 events have been detected successfully, one event was missed as in few frames two players are very close to each other, so it is hard to recognize possession. In our experiments, the detection of such events occurs if the event definition is met for an appropriate number of consecutive frames. In this very preliminary application of the proposed event detection framework, we referred to a heuristically chosen number of consecutive frames equal to 5. For example, if the distance between the ball and the player is very low for five consecutive frames, we have a Ball possession event.

Table 3. Event detection results

Detected Events	Total	detected	Miss	Accuracy
Ball Possession	14	13	1	92%
Kicking the ball	19	16	3	84%

In the next consecutive sequence of frames, if the distance between the ball and the player increases with respect to a specific threshold in an abrupt manner, we have a kicking the ball event. For kicking the ball event, 16 out of 19 events were detected successfully, three events were missed as in some cases it may happen that when the players kick the ball, the ball hits the next closest player in fewer than five frames.

In the future, based on the simple and low-level complex events, we are planning to detect more complex events such as "Pass the ball" and \Shot on goal" by effectively merging the simple and low-level complex events using logical and temporal operators. To define the higher-level complex events, we have taken into consideration events at different abstraction levels. To define the event "Pass the ball" let us consider Player1 and Player2 of the same team. While referring to players of the same team let us assume that the color of the upper half of the bounding box is the same. For instance, the higher-level complex event "Pass the ball" basically occurs if the following lower-level complex events occur. With respect to the successive sequence of frames, the event corresponding to "Player1 is in possession of the ball" is identified, if the distance between Player1 and the ball is very low for a few frames. Then, if the distance between Player1 and the ball increases in an appropriate number of the subsequent frames we can define the low-level complex event as "Kicking the ball". In the same consecutive sequence of frames if the distance between Player2 (of the same team as Player1) and the ball decreases up to a very low value and the possession of the ball is with Player2, while there is no other object between the ball and Player2, then we can define the higher-level complex event as "Pass the ball":

$$\left\langle 23, Pass, t+\tilde{k}, \langle passingPlayer, p_1\rangle, \langle receivingPlayer, p_2\rangle, \langle passedObject, ball\rangle \right\rangle$$

where 23 is the identifier, Pass is the event type, $t+\tilde{k}$ is time instance in which the event occurs. *passingPlayer* is the role performed by p1 on object ball, *receivingPlayer* is the role performed by p2 on object ball.

To define the event "Shot on goal" let us consider the three entities player, ball and goal post. The higher-level event “Shot on goal" basically occurs if, with reference to the consecutive sequence of frames, the player kicks the ball, the distance between the ball and the player increases and the distance between the ball and the goal post decreases up to a specific threshold. Then we can define the higher-level event as "Shot on goal":

$$\left\langle 20, ShotOnGoal, t+\tilde{k}, \langle KickingPlayer, p\rangle, \langle KickedObject, ball\rangle, \langle GoalPost, G\rangle \right\rangle$$

where 20 is the identifier, *ShotOnGoal* is the event type, $t+\tilde{k}$ is the event occurring instance, *KickingPlayer* is the role performed by *p*, *GoalPost* is the role of object *G*, when object ball approaches towards it.

7. CONCLUSIONS

In this paper, we have defined a few simple and complex events for the soccer video. We have also proposed a distance-based event detection system. The event detection system takes as an input bounding boxes associated with a confidence score for each object category. The system successfully detects the low-level complex events, such as: "Ball possession" and "Kicking the ball ". The results demonstrate the validity and the effectiveness of our methodology.

REFERENCES

[1] Challenges with image event processing, 2017. Poster DEBS 17.

[2] Adnan Akbar, Francois Carrez, Klaus Moessner, and Ahmed Zoha. Predicting complex events for pro-active iot applications. In Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, pages 327{332. IEEE, 2015.

[3] Adnan Akbar, Abdullah Khan, Francois Carrez, and Klaus Moessner. Predictive analytics for complex iot data streams. IEEE Internet of Things, 2017.

[4] Stefano Borgo and Riichiro Mizoguchi. A first-order formalization of event, object, process and role in yamato. In FOIS, pages 79-92, 2014.

[5] Pascual J Figueroa, Neucimar J Leite, and Ricardo ML Barros. Tracking soccer players aiming their kinematical motion analysis. Computer Vision and Image Understanding, 101(2):122-135, 2006

[6] Sebastian Gerke, Karsten Muller, and Ralf Schafer. Soccer jersey number recognition using convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 17-24, 2015.

[7] Ross Girshick. Fast r-cnn. arXiv preprint arXiv:1504.08083, 2015.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580-587, 2014

[9] Chung-Lin Huang, Huang-Chia Shih, and Chung-Yuan Chao. Semantic analysis of soccer video using dynamic bayesian network. IEEE Transactions on Multimedia, 8(4):749-760, 2006.

[10] Guoying Jin, Linmi Tao, and Guangyou Xu. Hidden markov model based events detection in soccer video. Image Analysis and Recognition, pages 605-612, 2004

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097-1105, 2012.

[12] Guang Li, Shubo Ma, and Yahong Han. Summarization-based video caption via deep neural networks. In Proceedings of the 23rd ACM international conference on Multimedia, pages 1191-1194. ACM, 2015.

[13] Daniel Link and Martin Hoernig. Individual ball possession in soccer. PloS one, 12(7):e0179953, 2017.

[14] Jia Liu, Xiaofeng Tong, Wenlong Li, Tao Wang, Yimin Zhang, and Hongqi Wang. Automatic player detection, labeling and tracking in broadcast soccer video. Pattern Recognition Letters, 30(2):103-113, 2009.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21-37. Springer, 2016.

[16] L. Pappalardo and P. Cintia. Quantifying the relation between performance and success in soccer. ArXiv e-prints, May 2017.

[17] Xueming Qian, Guizhong Liu, Huan Wang, Zhi Li, and Zhe Wang. Soccer video event detection by fusing middle level visual semantics of an event clip. In Pacific-Rim Conference on Multimedia, pages 439-451. Springer, 2010

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6):1137-1149, 2017.

[19] Wei Song and Hani Hagras. A big-bang big-crunch type-2 fuzzy logic based system for soccer video scene classification. In Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on, pages 2059{2066. IEEE, 2016.]

[20] P Thirumurugan and S Hasan Hussain. Event detection in videos using data mining techniques. International Journal of Computer Science and Information Technologies, 3(2):3473-3475, 2012.

[21] Dian W Tjondronegoro and Yi-Ping Phoebe Chen. Knowledge-discounted event detection in sports video. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40(5):1009-1024, 2010.

[22] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowd sourced video annotation. International Journal of Computer Vision, 101(1):184-204, 2013.

INTENTIONAL BLANK

REAP-SOS: A REQUIREMENT ENGINEERING APPROACH FORSYSTEM OF SYSTEMS

Felipe Lima Duarte[1] and Angélica Félix de Castro[2] and Paulo Gabriel Gadelha Queiroz³

[1]Center of Exact and Natural Science, UFERSA, Mossoró - RN, Brazil

ABSTRACT

A System of Systems (SoS) is a class of system composed of a set or arrangement of independent systems that together provide unique functionality for the end user. Due to its complexity, the Requirement Engineering (RE) process needs to undergo adaptations to fit the development of this type of system. In this context, the objective of this work is to propose a approach for the development of SoS, called REAP-SoS. The main characteristic of the REAP-SoS is the derivation of the individual missions and requirements of the constituent systems based on the general SoS assignments. In addition, the approach is also able to derive the requirements of the constituent systems of SoS. To validate the approach, a case study on a SoS urban traffic control and monitoring was performed.

KEYWORDS

System of Systems, Requirements Engineering, SysML

1. INTRODUCTION

A SoS is defined as the result of a set or arrangement of independent systems that are integrated and combined. The result of this union provides unique capabilities to users [13]. To distinguish SoS from complex and traditional systems, it is necessary to understand some of its characteristics. According to Maier [9], a SoS presents five main characteristics: operational independence, managerial independence, evolutionary development, emergent behaviour and geographical distribution.

In a software development process, one of the main objectives is to define the functionalities and constraints of the system [15]. These definitions are made in the Requirements Engineering (ER) process. According to Adu [1], it is responsible for providing a suitable mechanism to understand what clients want, to analyze their needs, the feasibility of what is requested, to specify, to validate and to manage the requirements.

A Systematic Review (SR) was carried out in 2016 with the purpose of identifying specific RE approaches applied to SoS, and twenty five approaches have been found. Some of them try to cover the entire development process, and others focusing on specific steps, such as specification or requirements management. Another aspect about the approaches founded is that some aim to identify the requirements of the SoS as a whole and thus to select existing systems that meet these

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 131–148, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80510

requirements, while others give greater importance in modelling the systems for a possible implementation.

In addition, as a result of SR, we found few studies with complete and well-documented methodologies, mainly regarding the design of the requirements of SoS and its Constituent Systems (CS). Many studies were limited to the SoS assignments or the requirements management activity.

Regarding traditional RE, existing approaches find it difficult to work in the context of complex systems, more specifically SoS, because of the requirements nature, which are fragmented, conflicting, unstable and often cannot be completely defined. Another issue is that, in SoS, there are two types of objectives, the objectives of the SoS as a whole; and also the objectives of individual systems [12].

Thus, the main purpose of this work is to present a specific approach to requirements engineering applied to SoS in such a way that it can guide the process of designing and modelling requirements. In addition, in order to validate the approach, we present a case study on a SoS for controlling and monitoring the urban traffic.

This work is organized as follows: section 2 presents related works; in section 3 presents the proposed approach; section 4 presents the case study and, finally, section 5 presents the final considerations of this work.

2. RELATED WORK

Some studies have already proposed specific RE approaches to the development of SoS, some of them are presented and discussed below.

The approach proposed by Holt et al. [7], is specific to SoS and was based on the traditional Context-Based Requirements Engineering (ACRE) approach, it contains three elements, which are:

- Ontology: used to describe concepts and terminologies of the application domain to be developed;

- Framework: defines a predefined set of system views, in practice, they are the artefacts to be created by the approach;

- Process: represented by a set of steps that are responsible for, on the basis of the ontology, generate the visions defined in the framework.

The approach proposes several steps in the process, among them, we can highlights: elicitation and development of requirements, verification and validation of requirements, control of requirements, traceability of requirements, monitoring of requirements, among others. We realized that this approach clearly represents what should be done to generate and manage the requirements, but does not offer many guidelines on how to do it.

Petrinca et al. [14] defines its approach as an iterative and top-down process, which begins with meeting stakeholder needs, represented by SoS missions. From these missions, a series of transformations in SysML models are carried out in order to derive the requirements of the Constituent Systems (CS). The process as a whole has been divided into three phases: definition of system context, definition of system behaviour and definition of architecture. It is an excellent approach for deriving the SOS missions and the requirements of the constituent systems, because it presents all the steps and transformations that must be made in the models generated by the approach. However, it does not address issues such as requirements management and validation. Another approach that deserves emphasis is defined by Lewis et al. [8], which propose a top-down approach, capable of understanding the requirements of the SoS as a whole, and bottom-up, capable of understanding the specificities of CS. It has the following phases: identifying the SoS context, identifying SoS and constituent objectives, understanding SoS interactions, identifying the capabilities and constraints of individual systems, and analyzing the gaps left by the process. It is a good approach to understanding the problem of defining and representing requirements in the context of SoS, but also does not offer many guidelines on how to do the steps described.

Other works that also address the theme of RE and SoS are: Ceccarelli et al. [5] and Yang-Turne et al. [21], which focuses mainly on the design and identification of requirements. Cavalcante et al. [4], which addresses the requirements modelling stage, proposing the use of implementations of the Goal Objective Requirements Engineering (GORE) approach, such as KAOS, i *, etc. Finally, the work of Vierhauser et al. [19] and Vierhauser et al. [20] addresses the requirements management stage, the latter proposing a monitoring approach based on three dimensions, namely: SoS requirements and SoS events.

From reading the works found in SR, which include the above works. With the exception of Petrinca et al. [14], the lack of a clear guide on how to generate the requirements of the constituent systems of SoS was perceived. Thus, the approach proposed in this work is intended to fill this gap left by other approaches. Regarding the approach proposed by Petrinca et al. [14], the main difference between the proposed approach and this work is the different activities of the process, such as modelling the structure, identify and model the missions of SoS, which we believe is easier to understand and use.

3. REAP-SOS: REQUIREMENTS ENGINEERING APPROACH FOR SOS

The REAP-SoS consists of a top-down proposal, which means that the requirements of CS are derived from the global SoS missions through iterations over SysML models. By missions, it is understood that they are typically viewed as goals, features, or a set of tasks. Missions can be related and contribute to the accomplishment of others, besides, as well as they can have a positive impact, they can also exert negative influences, making it impossible even to accomplish some task of the system. In SoS, there are two types of missions, which are: Individual Missions (IM), assigned to the constituent systems; and the Global Missions (GM), objectives assigned to SoS [17][18].

The goals of the proposed approach are:

- Analysis of SoS context and environment to be developed;
- Identify the CS of the SOS;
- Identification of the capabilities that are expected from the SoS as a whole, ie the general SoS;
- Transforming SoS missions into CS requirements;
- Modelling the SoS missions and CS requirements;
- Ensure the traceability of the requirements.

The approach, in addition to the identification of the SoS assignments and CS requirements, needs to address other aspects, such as: the type of SoS, the environment associated with SoS and the identification of interactions between systems. In addition to these, it is also important to understand the capabilities provided by individual systems.
The approach is composed of three elements, they are:

- Context Definition Phase: responsible for defining the entire context and environment to which SoS is embedded;
- Framework: responsible for defining the models and artefacts to be created by the approach;
- Conception and Modelling Phase: responsible for develop the elements of the framework.

With respect to traditional requirements engineering, the proposed approach differentiates itself by understanding from the beginning the complexity of the SoS assignments, thus trying to derive CS requirements through the global and individual SoS assignments. In addition, the process of identifying such systems, together with the implementation of a traceability study between requirements and constituent systems, are not characteristic of traditional approaches.

Figure 1 shows the REAP-SoS approach represented by the Business Process Model and Notation (BPMN) language. In the figure, we can see the iteration between the phases of context definition and conception and modelling phase. The latter is composed of some activities, which are: modelling the SoS structure, identifying and model the SoS missions, identifying, model and specify requirements, modelling system activities and modelling the system state. As can be seen from the BPMN loop notation, these activities are also performed iteratively and incrementally.

3.1. Context Definition Phase

The RAEP-SoS context definition is a phase responsible for identifying several aspects of the domain to which the SoS to be developed belongs. The first thing to be defined is whether the system to be specified really is a SoS. For this, the approach recommends identifying at least two CS that have the following characteristics: operational independence, managerial, emerging behaviour, evolutionary development and geographical distribution. In addition, such systems should contribute to a single specific purpose. The approach does not determine that all constituent systems should be identified, but it is important to try to find the majority of them because it will facilitate the next phases of the process.

The CS identification can be done by two main ways: by observing existing systems, trying to find some set of similar systems, analyzing their structure and their relationships. The other way is to apply questionnaires or perform interviews with experts on the problem that the SoS proposes to solve. In addition to these, Mokhtarpour et al. [10] propose an approach to CS selection for a SoS. Its methodology presents the following phases: identification of the missions, identification of candidate systems, selection of possible candidates, determination of alternatives and evolution of alternatives. Regardless of the manner used for identification. The approach recommends that for each system identified, a summary be made of it and describe its main objective in relation to SoS.

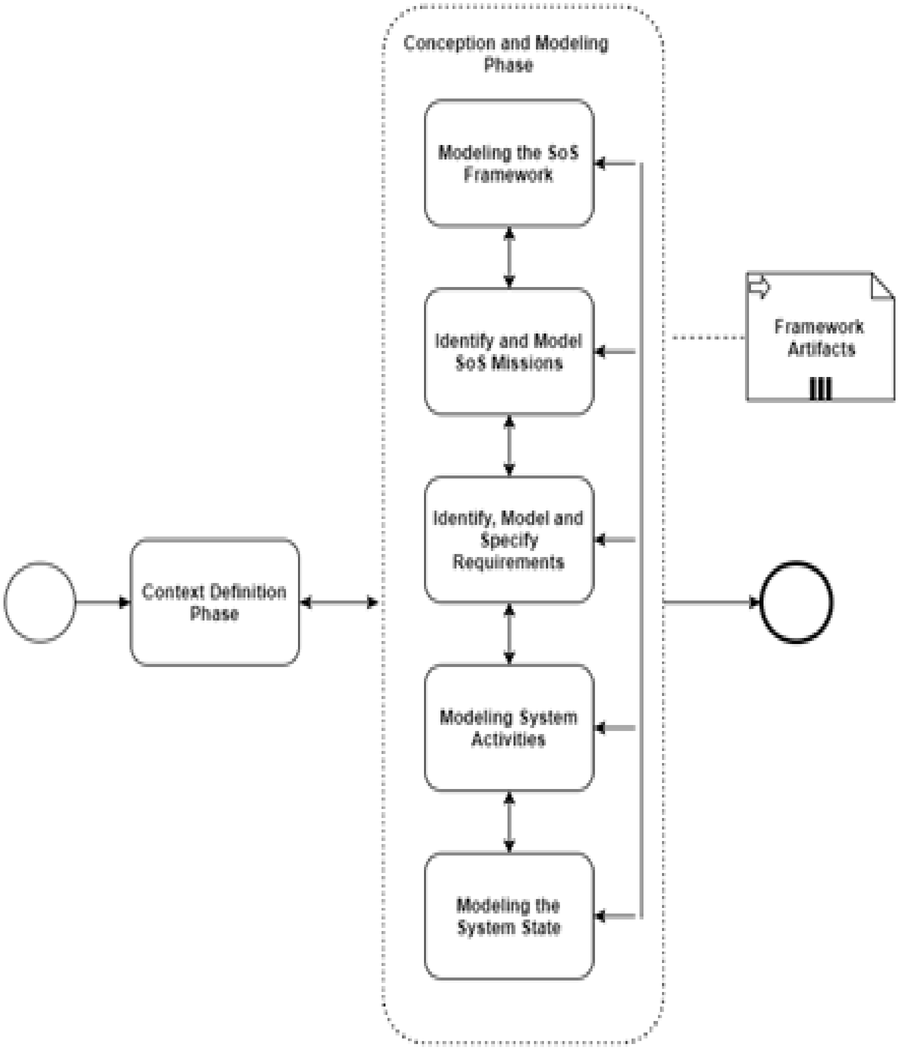

Conception and Modeling Phase
Modeling the SoS Framework
Identify and Model SoS Missions
Identify, Model and Specify Requirements
Modeling System Activities
Modeling the System State
Context Definition Phase
Framework Artifacts

Figure 1. REAP-SoS Approach in BPMN

After define if the system really is a SoS, the approach recommends identifying the environment to which the SoS is inserted, because each of the constituent systems of a SoS are influenced directly by its environment, the aspects of the environment to be defined are:

- Entity: is all that can influence SoS, for example: technologies, approaches, people, companies, stakeholders and others. It is important to emphasize that the concept of entity differs somewhat from the concept of stakeholders, since they do not necessarily have an interest in the project, but can only positively or negatively influence the SoS or the constituent systems;

- Influence: entities can and often exert different levels of influence between the constituent systems and SoS. A mapping of this influence is important to analyze and assist in prioritizing the changes in SoS and its systems.

Understanding these two aspects is important in anticipating future changes to the requirements of SoS as a whole and individual systems. The approach recommends the use of three types of influence, which are: low (L), medium (M) or high (H). The use of only three levels is due to the attempt of the approach not to generate excessive complexity. Another recommendation of the approach is to verify whether the entity can be seen as a stakeholder of a CS. If it is, it will have a greater degree of influence in this system and smaller in the other CS. If the entity cannot be seen as a stakeholder of any CS, it will probably have a greater degree of influence on the SoS as a whole.

To finalize the SoS context phase, the REAP-SoS recommends the creation of a feasibility study, to ensure that the SoS to be developed:

- Contributes to the organization's objectives;
- Can be built with the technology and staff available by the organization;
- Can be integrated with systems already present in the organization.

3.2. Framework

The framework is the element of the approach responsible for the formal definition of the artefacts that must be created by the conception and modelling phase, which are:

- Block definition diagram: responsible for modelling the SoS structure, presenting its CS;
- Block definition diagram (Missions): responsible for modelling the global and individual missions of SoS;
- Requirements diagram: responsible for representing the requirements of CS;
- Activity diagram: responsible for providing the dynamic structure of the systems, responsible for showing the flow of activities. showing how an activity depends on one another;
- State machine diagram: responsible for showing the possible states of an CS and the transactions responsible for its state changes.

3.3. Conception and Modelling Phase

The conception and modelling phase consists of a set of activities whose main purpose is to conceive and model CS requirements. It is important to note that they can and should be done in an iterative and incremental way. In addition, each activity is responsible for generating a framework artefact, as can be seen in Figure 2. The activities of this phase are: modelling the SoS structure, identifying and modelling SoS missions, identifying and modelling the requirements, model the static structure of the system and map the requirements. The main objective is to generate the artefacts required by the framework.

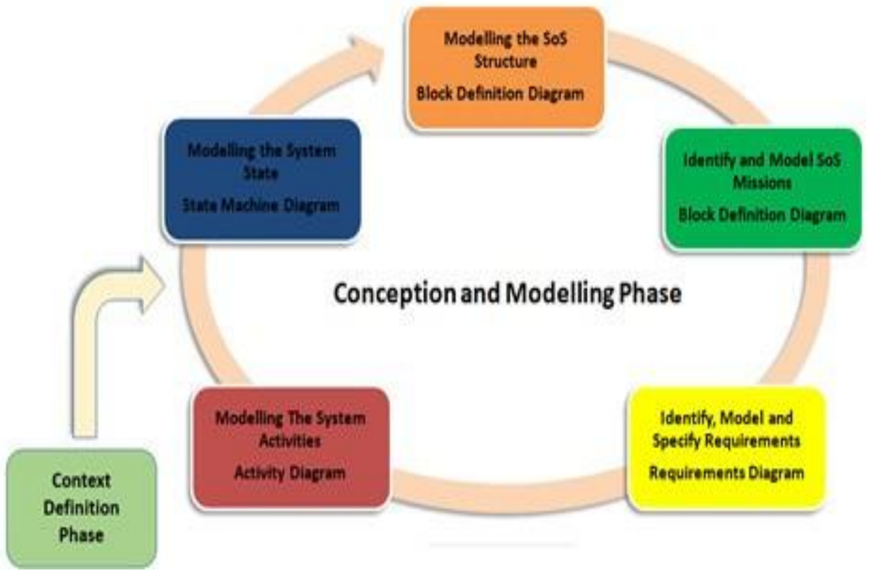

Modelling the SoS Structure
Block Definition Diagram
Identify and Model SoS Missions
Block Definition Diagram
Identify, Model and Specify Requirements
Requirements Diagram
Modelling The System Activities
Activity Diagram
Modelling the System State
State Machine Diagram
Conception and Modelling Phase
Context Definition Phase

Figure 2. Activities of the Conception and Modelling Phase

3.3.1 Modelling the SoS Structure

The purpose of this activity is to identify and formalize the SoS structure, identifying the CS which are part of SoS. It is also important to note that one SoS can be part of another SoS, this must also be modelled.

To model the general structure of SoS, the approach recommends that, given the constituent systems identified in the SoS context, model a SysML block definition diagram containing the main SoS systems identified so far and how they are related. It is important to make it clear that this is an early version of the diagram and that it can be changed after building other artefacts.

The approach also recommends inserting the <<SoS>> stereotype to identify the blocks as a system of systems, in addition to the stereotype <<CS>> to identify the block as a SoS constituent system.

The following is a summary of this activity, defining inputs, execution and output:

- Inputs: constituent systems identified in the context definition phase;
- Execution: create and model a block diagram to represent the SoS structure;
- Outputs: SoS structure (SoS and CS).

3.3.2 Identify and Model SoS Missions

The purpose of this activity is to identify and model the SoS and CS missions. For this, starting from the previous model, the blocks must be separated and for each of them their missions must be identified. If the block has the SoS stereotype, the missions identified will be global missions, if the block has the CS stereotype, the missions modelled will be individual missions of the system. The REAP-SoS approach then recommends techniques for identifying and modelling these missions, these techniques should be assisted by the entities defined in the context definition phase.

The REAP-SoS approach does not determine the techniques for identifying the SoS missions, instead it leaves the user free to choose the best form. However, the approach recommends two techniques, which are: personas and task analysis.

Due to the complexity of the system and the possibility of a large number of stakeholders from different domains, the approach recommends the use of a technique called personas. This technique assists the system users in understanding their characteristics, needs and objectives, thus supporting requirements engineering [3]. The personas technique mainly consists of collecting data about users, obtaining an understanding of their characteristics and defining descriptions for groups of users. Based on this understanding, focus on these personas throughout the software creation process [16]. The objective of using this technique is to have a real understanding of the different stakeholders of the various systems that make up the SoS, thus helping the requirements elicitation process.

Another technique is task analysis, it is a top-down approach, in which tasks of a higher abstraction degree are decomposed into sub-tasks and eventually detailed into events that describe it. The main goal of this approach is to build a hierarchy of tasks [22]. It can be observed that this concept fits well into the definition of the SoS missions, since the global missions are broken down into individual missions that carry them out, thus forming a hierarchy of missions.

To Model the SoS missions, the approach recommends the use of the SysML block definition diagram, since it is a general purpose diagram, there is no problem in using it more than once, in addition, since all other diagrams recommended by the REAP-SoS approach are made in SysML, it is interesting to model the missions also with this language. The difference here is that if the missions were derived from a block with stereotype <<SoS>>, then the missions must have the <<Global Mission>> stereotype, if they are derived from CS, they should have the <<Individual Mission>> stereotype. Besides this information the approach also recommends that the block (SoS or CS) that originated the mission be placed in stereotype form. It is observed that a tree structure will form and probably the leaves of the trees will be the individual missions of the constituent systems.

A summary of this activity is:

- Inputs: SoS structure (SysML block definition diagram) and entities of the context definition phase;
- Execution: for each block of the SoS structure, create and model their missions, recommended techniques: personas, task analysis;
- Outputs: SoS and CS missions (SysML block definition diagram).

3.3.3 Identify, Model and Specify Requirements

The objective of this activity is to identify the main functionalities of the systems that will be responsible for carrying out the SoS mission. These should have a high degree of abstraction, characterizing them as CS user requirements. For this, starting from the previous model, for each identified individual mission, the requirements that fulfil this mission must be derived.

In order to carry out, as in the previous phase, stakeholders (entities) should be consulted. As it is CS, more traditional techniques are recommended, such as: interviews, questionnaires, ethnography and brainstorming. For requirements modelling, the approach recommends the use of SysML requirements diagram.

A summary of this activity is:

- Inputs: SoS and CS missions (SysML block definition diagram);
- Execution: for each mission, to create and model the requirements that fulfil them, recommended techniques: interviews, questionnaires, ethnography and brainstorming;
- Outputs: CS requirements (SysML requirements diagram).

3.3.4. Modelling the System Activities

The main objective is the specification of the behaviour of the SoS as a whole and how the collaboration of CS occurs, from the functional point of view. In addition, specify what will be subsequently designed, or directly constructed, thereby reducing the abstraction level of the scope, making it easier to understand what has to be done by the developers.

A summary of this activity is:

- Inputs: SoS structure (SysML block definition diagram) and CS requirements (SysML requirements diagram);
- Execution: create and model the activities of the systems that make up the SoS;
- Outputs: SoS and CS activities (SysML activity diagram).

3.3.5 Modelling the System State

The objective of this activity is to provide a better understanding of the CS and to facilitate the modelling of this system for the project team. This is done by creating state diagrams of the constituent systems.

A summary of this activity is:

- Inputs: SoS structure (SysML block definition diagram) and CS requirements (SysML requirements diagram);
- Execution: if the states of the systems can be easily identified, create and model its states;
- Outputs: CS states (SysML state diagram).

4. CASE STUDY

The present case study aims to present the use of the REAP-SoS approach. It consists of designing and modelling the requirements of a SoS for controlling and monitoring urban traffic.

Nowadays, heavy traffic and the increase in the number of traffic accidents have caused a high cost (economic, social and environmental) for companies. In Brazil, especially in large metropolises, public managers have one of their main challenges in urban mobility. Several alternatives have been and are made to solve this problem, among them, we can mention: improvement of infrastructure, traffic restriction program (To combat air pollution and traffic jam the city only allows vehicles whose licence numbers end with certain digits to drive on particular weekdays.) and collection of fees for urban areas. However, they did not have the desired effect [2].

The use of technologies in transport and vehicle infrastructure results in so-called Intelligent Transport Systems (ITS) [6]. These systems have, among other objectives, to provide efficient telecommunication and computer solutions for urban traffic problems. Among the objectives of these systems can be highlighted: traffic control and traffic lights, management of emergency services, automatic collection of tariffs in collective transportation, among others.

The context of ITS is directly related to a relevant theme, which is the "Smart Cities", which can be seen as a set of steps that a citizen takes, together with services and technologies to make the city a more habitable environment/comfortable, with more efficient services and ready to suit any situation. Among the essential aspects for the creation of a smart city, it is necessary a great modern and intelligent digital infrastructure able to meet the demands that these cities need [11].

4.1. Context of Urban Monitoring and Control

In this section, as determined by the REAP-SoS approach (Section 3.1), the CS, the SoS existence check, the SoS type definition, as well as the SC entities for the urban traffic monitoring and control system are defined and presented. All information, as well as the missions defined subsequently, were taken from the observation and reading of the work related to the SIMTUR project, which can be accessed at the following address: *http://projeto.unisinos.br/simtur*. The following are the CS identified:

- Smart Cars (CS1): this type of system, in addition to all the functionalities of a conventional car, would also be the primary agent responsible for collecting information about traffic and sending it to the traffic control system. Based on the information collected in real time, the decisions about the best routes and the following routes would also be taken in real time;

- Smart buses (CS2): many people rely on collective transportation systems to move around in big cities, so, in addition to smart cars, it is also very important to have intelligent public transport systems. These, in addition to all the functionalities of smart cars, with regard to improving urban traffic, would also be responsible for collecting and reporting data such as: route performed on the day, location, average time at each stop or terminal, as well as other information;

- Waze (CS3): it is known that cars and buses with the features described above are practically unviable nowadays, from an economic point of view. For this, a cheaper alternative would be the use of applications such as *Waze* (*https://www.waze.com*) or similar (*Google Maps*, *Tom Tom Go*). This application would be responsible for sending the vehicles location to the traffic control system;

- Smart traffic light (CS4): based on information collected in real time, the intelligent traffic lights would be the main actuators in the control flow of the main roads in the cities, having as main objective the control of the time of opening and closing based on the information of the most congested routes;

- Traffic controller (CS5): this system would be the main driver of urban traffic in order to reduce congestion. It would be responsible for receiving information from intelligent vehicles, identifying possible bottlenecks and triggering intelligent traffic lights;

- Public transportation application (CS6): this system is responsible for providing the user with information about bus stations, routes, which bus is closest, how long to reach the bus stop, among others.

After identifying the systems, it is necessary to verify if they have the characteristics of a SoS. The following is a brief summary of two of the five systems identified (smart cars and smart traffic lights).

In the case of smart cars, the main features that characterize it as a SoS constituent system are:

- Operational Independence: all cars can be seen as a system that operates without the need for other systems, for example: a car operates without the need for other vehicles to be running;

- Management Independence: all the functionalities present in cars, be they related to locomotion or information collection are controlled by the vehicle itself, without the interference of other systems;

- Geographic distribution: each vehicle occupies and operates in a space geographically distant from other systems;

- Emerging Behaviour: each car can collect and pass on information to both the central control units and other vehicles.

In the case of intelligent traffic lights, the main aspects that characterize it as a constituent system of SoS are:

- Operational Independence: a semaphore shall operate independently of the operation of other traffic lights or vehicles and systems;

- Management Independence: each traffic light can be managed independently of other traffic lights. Although, the information obtained by the control system will affect its management;

- Geographic distribution: each traffic light will be geographically separated by city roads;

- Emerging Behaviour: control the flow of cars, improving congestion control.

After identified the CS and ensuring that the system really is a SoS, the next step is to define the entities and the degree of influence they have on systems, the entities identified are

- Drivers: responsible for driving vehicles on the roads of the city;

- Users of public transport: users of collective transportation systems such as buses, subways etc;

- Pedestrians: although they do not have a direct influence on the problems that the SoS propose to solve, they are part of the organization chart of the traffic in the big cities;

- Technology: the entity that makes everything possible, since all the solutions go through the development of a great technological apparatus;

- Users of Traffic Control System (TCS): users of traffic control systems. Although this system can be autonomous, there must be users to manage and control some aspects of this system;

- Users of the Public Transport Control System (PTCS): as in the previous entity, there may also be users in the public transport control system;

- Local Government: probably the financier of the project, since it is the function and duty of public managers to solve the problems of mobility in their cities;

- Public transportation concessionaire company: in Brazil, most of the transportation companies are private and have partnerships with the government, and can exploit that activity over a period of time. Being part of the solution directly, it is also an entity interested in the SoS project.

Regarding the influence that these entities exert on the systems, in Table 1, a summary is presented. Following the approach, three degrees of influence were defined: low (L), medium (M) or high (H). Entities such as: drivers and users of public transport, in addition to the degree of influence in systems such as automobiles and public transport, also have a high degree of influence in the SoS as a whole, since they are the main stakeholders in SoS.

Table 1. Levels of Influence of Entities.

Entities	**CS1**	**CS2**	**CS3**	**CS4**	**CS5**	**SoS**
Drivers	H	L	M	L	L	H
Users of public transport	L	H	M	L	L	H
Pedestrians	L	L	H	L	L	L
Technology	H	H	H	H	H	H
Users of TCS	M	L	H	H	M	H
Users of PTCS	L	M	M	L	H	M
Local Government	M	M	H	H	H	H
Public transportation concessionaire	L	H	M	L	M	M

To conclude, the approach recommends conducting a feasibility study. It aims to assess whether the system can actually be built, analyzing aspects such as: available technology, time, personnel and cost.

As this case study is only intended to present and validate a concept, as well as to deepen the knowledge of designing and modelling the requirements of a SoS using REAP-SoS. The feasibility study was not created.

4.2. Modelling the SoS Structure

As recommended by the REAP-SoS approach (section 3.3.1), the SoS structure for urban traffic control and monitoring is presented, as can be seen in Figure 3. The stereotype <<SoS>> is placed to identify the blocks as a system of systems, and the stereotype <<CS>> is inserted to identify the block as a SoS constituent system.

The main SoS is the *Traffic Control and Monitoring System*, which consists of two other SoS, which are the *Congestion Control System* and *The Public Transport Information System*. For once, these SoS are composed of several CS, which are the *Traffic Controller,* the *Smart Traffic Light*, the *Smart Bus*, among others. It is also important to note that the *Information Capture System* (CS) can be instantiated either by *Waze* or *smart car* or *Smart Bus*, the latter, beside participate in the *Information Capture System*, also participates in the *Public Transport Information System*.

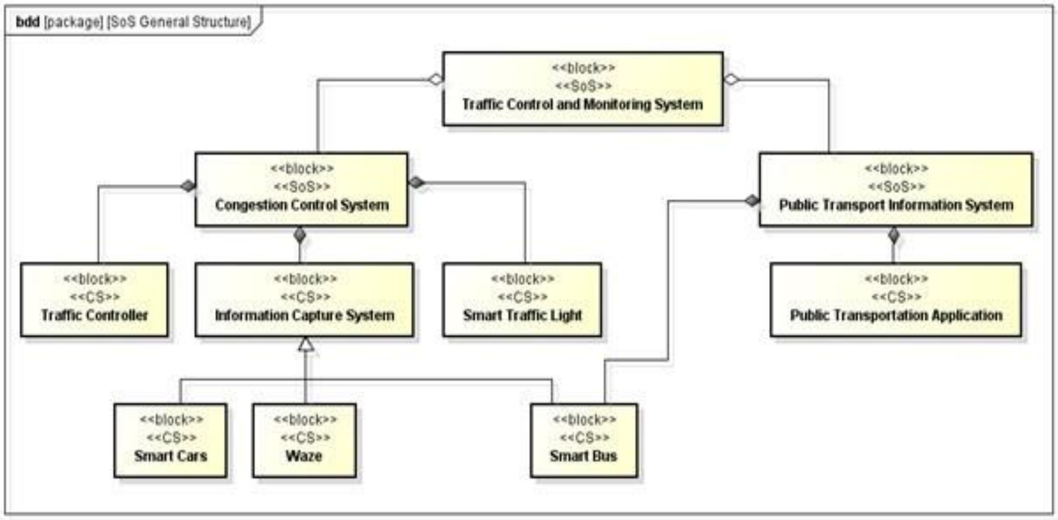
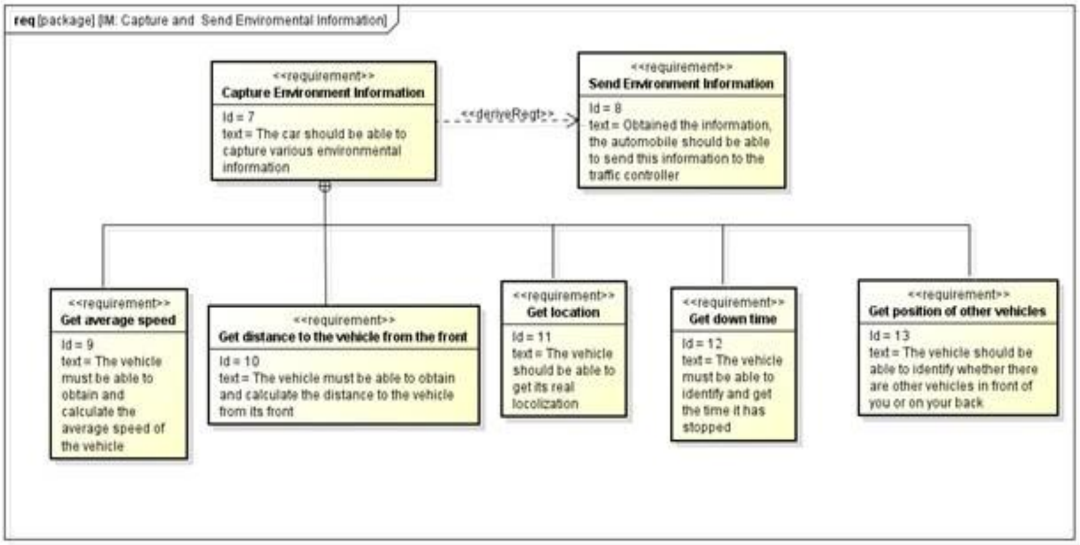


Figure 3. SoS general structure

4.3. Identify and Model SoS Missions

Following the REAP-SoS approach (section 3.3.2). The global and individual missions of SoS and CS have been defined and modelled. As can be seen in Figure 4, each block is represented by a mission, the stereotypes determine whether the mission is global or individual. In addition, a stereotype of the block (system) that originated the mission is added.


req [package] [IM: Capture and Send Enviromental Information]
<<requirement>>
Capture Environment Information
<<deriveReqt>>
<<requirement>>
Send Environment Information
<<requirement>>
Get average speed
<<requirement>>
Get distance to the vehicle from the front
<<requirement>>
Get location
<<requirement>>
Get down time
<<requirement>>
Get position of other vehicles

Id = 7
text = The car should be able to capture various environmental information

Id = 8
text = Obtained the information, the automobile should be able to send this information to the traffic controller

Id = 9
text = The vehicle must be able to obtain and calculate the average speed of the vehicle

Id = 10
text = The vehicle must be able to obtain and calculate the distance to the vehicle from its front

Id = 11
text = The vehicle should be able to get its real locolization

Id = 12
text = The vehicle must be able to identify and get the time it has stopped

Id = 13
text = The vehicle should be able to identify whether there are other vehicles in front of you or on your back

Figure 4. SoS Missions

The global missions identified are: managing and monitoring urban traffic, controlling traffic congestion, and monitoring and reporting the status of public transportation. The individual missions identified are: identifying congestion locations, controlling vehicle flow, capturing and sending information from the environment, and finally, consulting itineraries and bus stops.

4.4. Identify and Model Requirements

As recommended by the REAP-SoS approach (section 3.3.3), after the modelling of the missions, the requirements diagram was created. As can be seen in Figure 5, the requirements that accomplish the individual mission *Capture and Send Environmental Information* are presented. In this model, two main requirements are presented, *Capture Environment Information* and *Send Environment Information*, the latter, to be performed accurately That the former is implemented, as can be seen in the notation <<*deriveRect*>>. The *Capture Environment Information* requirement consists of several other requirements, such as: *average speed*, *time stopped*, *identifying other vehicles* and so on. They are all responsible for capturing a type of environmental information, such as: average speed of the vehicle, identification of other vehicles on the road, vehicle stopped time etc.

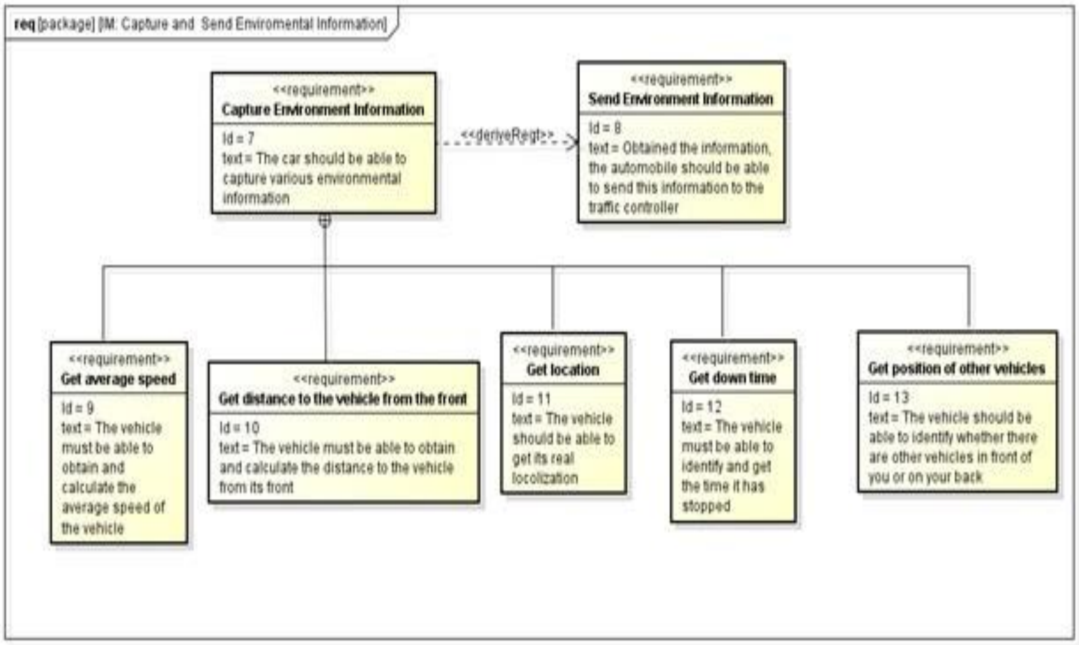

req [package] [IM: Capture and Send Enviromental Information]
<<requirement>>
Capture Environment Information
Id = 7
text = The car should be able to capture various environmental information
<<deriveReqt>>
<<requirement>>
Send Environment Information
Id = 8
text = Obtained the information, the automobile should be able to send this information to the traffic controller
<<requirement>>
Get average speed
Id = 9
text = The vehicle must be able to obtain and calculate the average speed of the vehicle
<<requirement>>
Get distance to the vehicle from the front
Id = 10
text = The vehicle must be able to obtain and calculate the distance to the vehicle from its front
<<requirement>>
Get location
Id = 11
text = The vehicle should be able to get its real locolization
<<requirement>>
Get down time
Id = 12
text = The vehicle must be able to identify and get the time it has stopped
<<requirement>>
Get position of other vehicles
Id = 13
text = The vehicle should be able to identify whether there are other vehicles in front of you or on your back

Figure 5. Smart Car Requirement Diagram

4.5. Modelling Activities of the Constituent Systems

As recommended by the REAP-SoS approach (section 3.3.4), an activity diagram was created for the *Congestion Control System*, as can be seen in Figure 6. The intention is to present the flow of information between the constituent systems of this SoS. *Smart cars*, *Smart buses* and *Waze* get the necessary information, send this information to *Traffic Controller*, which is responsible to processes, identifies congestion and sends this information to the *Traffic Lights*, which in case of congestion changes its configuration.

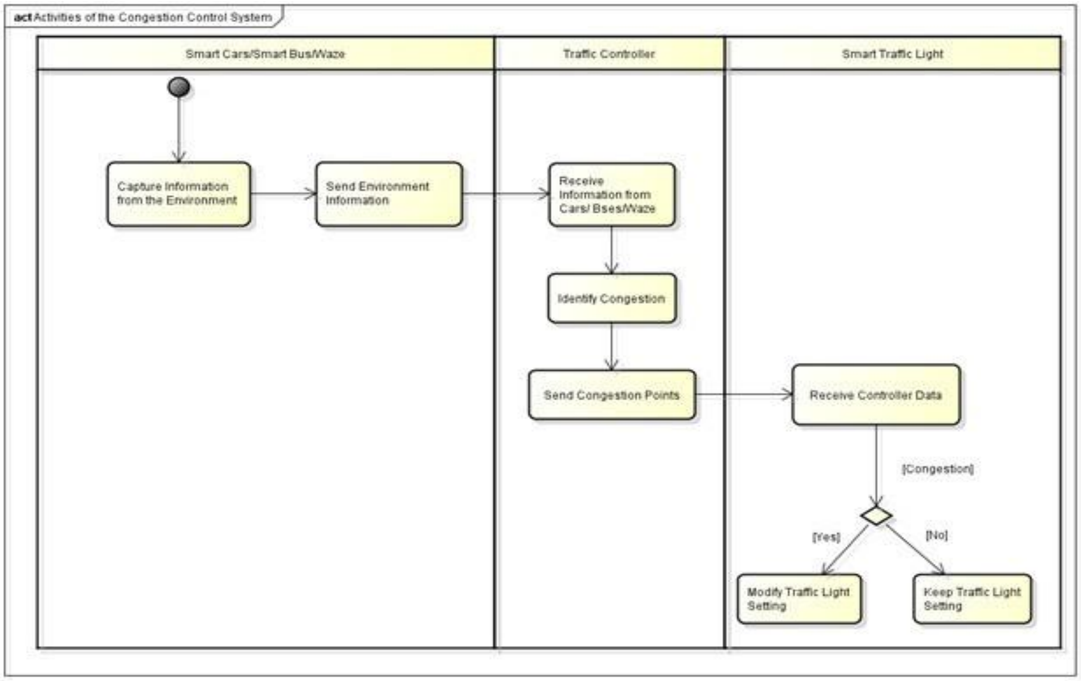

act Activities of the Congestion Control System
Smart Cars/Smart Bus/Waze
Traffic Controller
Smart Traffic Light
Capture Information from the Environment
Send Environment Information
Receive Information from Cars/ Bses/Waze
Identify Congestion
Send Congestion Points
Receive Controller Data
[Congestion]
[Yes]
[No]
Modify Traffic Light Setting
Keep Traffic Light Setting

Figure 6. Activities of Control Vehicle Congestion SoS

4.6. Modelling the System State

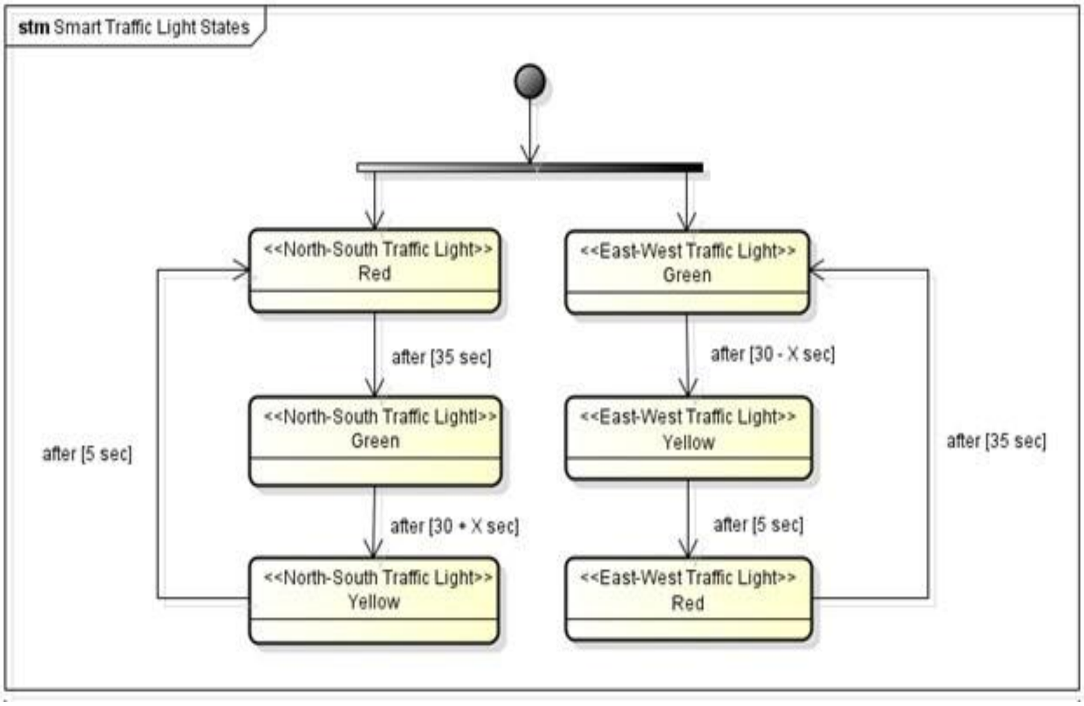

stm Smart Traffic Light States
<<North-South Traffic Light>> Red
after [35 sec]
<<North-South Traffic Lightl>> Green
after [30 + X sec]
<<North-South Traffic Light>> Yellow
after [5 sec]
<<East-West Traffic Light>> Green
after [30 - X sec]
<<East-West Traffic Light>> Yellow
after [5 sec]
<<East-West Traffic Light>> Red
after [35 sec]

Figure 7. States of Smart Traffic Light System

As discussed in section 3.3.5, the *Smart Traffic Lights* state diagram was created to show the operation of this system. As can be seen in Figure 7, there are two modelled traffic lights, which represent the street with north-south direction and the other that represents the street with east-west direction, the stereotype is identifying which semaphore the state belongs to. Three states were defined: *Green*, *Yellow* and *Red*. The main difference from this traffic light to the conventional one is the use of Variable X in the transition from green to yellow state (both signals), note that while it decreases the time in a traffic light, it increases the time in another, that variable will allow to modify the configuration of the traffic light and to provide a longer stay from the green to the more congested street.

5. CONCLUSIONS

This paper presented the REAP-SoS approach, which is an RE approach applied to the development of SoS. It is composed of three elements, they are: the context of the SoS, responsible for the definition of the environment to which the SoS is inserted; Design and modelling, responsible for describing the stages responsible for generating the SOS missions and the CS requirements; and finally; The framework, responsible for the formal definition of the artefacts that must be created by the process.

The proposed approach was based on other approaches specific to the development of SoS. Reap-SoS tries to unite the best of these methodologies, besides inserting and modifying elements in order to obtain a robust and complete approach, mainly in the design and modelling stage of the Requirements of SoS as a whole and CS.

In addition, in order to validate the approach, a case study on a SoS for control and monitoring of urban traffic was carried out. This system main purpose is to improve the flow of vehicles, reducing congestion and facilitating the life of the population of big cities. As it is a complex system with many CS, the use of a specific approach to SoS was necessary.

We believed the Reap-SoS can contribute to the subject of study, as well as serve as a basis for others approaches. In addition, it is expected to be the beginning of a comprehensive approach that will encompass all stages of the development of a SoS. Thus, as a starting point, it is expected that in the future, we also address architecture design, implementation, testing and management of SoS.

REFERENCES

[1] Adu, Michael (2014) “Inadequate Requirements Engineering Process: A Key Factor for Poor Software Development in Developing Nations: A Case Study”, International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol 8.

[2] Buarque, S. C. (2008) “Construindo o desenvolvimento local sustentável”, Garamond.

[3] Castro, John W. & Acuña, Silvia T. & Juristo Juzgado, N. (2008) “Enriching requirements analysis with the personas technique”, First Workshop on the Interplay between Usability Evaluation and Software Development.

[4] Cavalcante E. & Batista T. & Bencomo N. & Sawyer P. (2015) “Revisiting Goal-Oriented Models for Self-Aware Systems-of-Systems”, IEEE International Conference on Autonomic Computing, 231—234.

[5] Ceccarelli A. & Mori M. & Lollini P. & Bondavalli A. (2015) “Introducing Meta-Requirements for Describing System of Systems”, IEEE 16th International Symposium on High Assurance Systems Engineering, 150—157.

[6] Colombo, A. & Del Vecchio, D. (2012) “Efficient Algorithms for Collision Avoidance at Intersections”, 15th ACM international conference on Hybrid Systems: Computation and Control, pp. 145—154.

[7] Holt J. & Perry S. & Payne R. & Bryans J. & Hallerstede S. & Hansen F. O. (2015) “A Model-Based Approach for Requirements Engineering for Systems of Systems”, IEEE Systems Journal, 252—262.

[8] Lewis G. A. & Morris E. & Place P. & Simanta S. & Smith D. B. (2009) “Requirements Engineering for Systems of Systems”, 3rd Annual IEEE Systems Conference, 247—252.

[9] Maier, M.W. (1996) “Architecting principles for systems-of-systems.” In INCOSE International Symposium 6 (1):,565-573.

[10] Mokhtarpour B. & Stracener J. (2014) “A Conceptual Methodology for Selecting the Preferred System of Systems”, IEEE Systems Journal, 1—7.

[11] Monzon, A. (2015) “Smart cities concept and challenges: Bases for the assessment of smart city projects”, Smart Cities, Green Technologies, and Intelligent Transport Systems, Springer, Cham, pp. 17—31.

[12] Ncube C, (2011) “On the Engineering of Systems of Systems: key challenges for the requirements engineering community”,{Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems, 70—73.

[13] OUSD(AT&L), DoD. (2008) “Systems and Software Engineering. Systems Engineering Guide for Systems of Systems”. Technical Report Version 1.0. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Department of Defense.

[14] Petrinca P. & Gammaldi M. & Tirone L (2012) “A SysML-based Approach for the Specification of Complex Systems”, INCOSE International Symposium, 713—786.

[15] Ribeiro, L. C. M. & Ramos, C. S. & Brito, M. F. & Figueiredo, R. M. C. (2011). Definição de um processo de engenharia de requisitos para software embarcado na industria automotiva baseada em uma arquitetura de processos de software. In Workshop Anual do MPS, Campinas.

[16] Schneidewind, L. & Hörold, S. & Mayas, C. & Krömker, H. & Falke, S. & Pucklitsch, T. (2012) “How Personas Support Requirements Engineering”, Usability and Accessibility Focused Requirements Engineering (UsARE), 2012 First International Workshop, 1—5.

[17] Silva, E. & Batista, T. & Oquendo, F. (2015) “A Mission-Oriented Approach for Designing System-of-Systems”, 2015 10th System of Systems Engineering Conference (SoSE).

[18] Silva, E. & Cavalcante, T. & Batista, F. & Oquendo, F. & Delicato, C. & Pires, P. F. (2014) “On the characterization of missions of systems-of-systems”,Proceedings of the 2014 European Conference on Software Architecture Workshops. New York, NY, USA: ACM.

[19] Vierhauser M. & Grünbacher P. (2014) “A Requirements Monitoring Infrastructure for Systems of Systems”, ASE '14, 887—890.

[20] Vierhauser M. & Rabiser R. & Grünbacher P. & Aumayr B. (2015) “A Requirements Monitoring Model for Systems of Systems”, IEEE 23rd International Requirements Engineering Conference (RE), 96—105.

[21] Yang-Turner F., & Lau L. (2011) “A Pragmatic Strategy for Creative Requirements Elicitation: From current work practice to future work practice”, Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems, 28—31.

[22] Zowghi, D. & Coulin, C. (2005) “Requirements Elicitation: A Survey of Techniques, Approaches, and Tools”, Engineering and Managing Software Requirements, Springer, 19—46.

AUTHORS

Felipe Lima Duarte

Information Technology Analyst of the Information and Communication Technology Superintendence (ICTS) of the Federal Rural Semiarid University (UFERSA). He holds a degree in Computer Science from the Federal Rural Semiarid University (UFERSA) and currently holds a Master's Degree in Computer Science by UFERSA.

Angélica Félix de Castro

Graduated in Computer Science from the Federal University of Rio Grande do Norte (2000), Master in Geodynamics from the Federal University of Rio Grande do Norte (2002), PhD in Geodynamics from the Federal University of Rio Grande do Norte and Christian-Albrecht Universitat zu Kiel, Germany (2007) and Post-Doctorate in Computing from the University of Bristol, England (2015). She worked as a teacher of the Higher Magisterium at the Federal Institute of Bahia (IFBA), Campus of Vitória da Conquista, from 2006 to 2008 and is currently Associate Professor I at the Federal Rural Semi-Arid University.

Paulo Gabriel Gadelha Queiroz

He holds a degree in Computing from the Federal University of Ceará (2007), a master's degree (2009) and a doctorate (2015) from the University of São Paulo (ICMC-USP). He is currently Assistant Professor I of the undergraduate course in Computer Science at the Federal Rural Semi-Arid University (UFERSA). Has experience in the area of Computer Science, with emphasis on Software Engineering. The major research interests are: reuse, product line, Web systems, Web Services, application generators and critical embedded systems.

REAL TIME EMULATION OF PARAMETRIC GUITAR TUBE AMPLIFIER WITH LONG SHORT TERM MEMORY NEURAL NETWORK

Thomas Schmitz and Jean-Jacques Embrechts

Department of Electrical Engineering and Computer Science, Liege University,
Montefiore Institute, Belgium

ABSTRACT

Numerous audio systems for musicians are expensive and bulky. Therefore, it could be advantageous to model them and to replace them by computer emulation. In guitar players' world, audio systems could have a desirable nonlinear behavior (distortion effects). It is thus difficult to find a simple model to emulate them in real time. Volterra series model and its subclass are usual ways to model nonlinear systems. Unfortunately, these systems are difficult to identify in an analytic way. In this paper we propose to take advantage of the new progress made in neural networks to emulate them in real time. We show that an accurate emulation can be reached with less than 1% of root mean square error between the signal coming from a tube amplifier and the output of the neural network. Moreover, the research has been extended to model the Gain parameter of the amplifier.

KEYWORDS

Tube Amplifiers, Nonlinear Systems, Neural-Network, Real-Time.

1. INTRODUCTION

The modeling of nonlinear systems has been a central topic in many engineering areas, as most real-world devices exhibit nonlinear behaviors. In particular, the study of distortion effects for guitar players has been largely covered [1, 2, 3]. The reason is that musicians like the sound of tube amplifiers (in which each amplifier stage is composed of old vacuum-tube triodes). Guitarists define the sounds as more dynamic and warmer than those provided by solid state amplifiers (full transistors amplifiers). However, the tube amplifiers are often bulkier, more expensive, heavier and more fragile. This explains the large interest of the musician community for computer emulations. Even if musicians agree that these emulations get better with age, no exact correspondence between the sound coming from a tube amplifier and its emulation has been found in the literature.

In previous researches we have focused on Volterra series models [4, 5] and more specially on its subclass, the Wiener-Hammerstein cascade models [6, 7]. More specifically, researches on Hammerstein model have led to a fast Hammerstein Identification by Sine Sweep (HKISS) method [1, 2]. However, this kind of model is not sufficiently complex to correctly perform the emulation of wide range of guitar signals [1]. In this paper we propose to take advantage of the new progress made in the field of neural-networks (NN) and to evaluate the possibility of performing an accurate emulation of the *ENGL Retro Tube 50* amplifier in real time (RT).

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 149–157, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80511

The paper is organized as follows: the neural-network used to emulate the amplifier is presented in section 2. In section 3, the learning method and the data-set pre-processing method are described. In Section 4, the sound of the real system (i.e. the tube amplifier) and the sound from the emulated system (i.e. the NN) when a guitar signal is provided at the input are compared. Section 5 explains how to extend this model to include the amplifier's parameters (gain, equalization, ...). In this paper, the *Gain* parameter is taken as example. The effect of the knob *Gain* is to add more and more musical distortion to the guitar signal.

2. RECURRENT NEURAL-NETWORKS

Recurrent Neural-Networks (RNN) seem well suited to learn the nonlinear behavior of a tube amplifier. As the nonlinearities can change according to the input frequencies, it seems natural to take the previous values of the input signal *x* into account (the signal coming from the guitar) in order to compute the corresponding output signal *pred (the signal that emulate the output signal of the tube amplifier)* as depicted at Fig.1. In this case, to calculate the prediction *pred[n]* of the system, the RNN has to be fetch with a sequence of the last N values of the input signal *[x[n-(N-1)], ..., x[n]]*, where *N* is called the number of time steps (*num_step*). One can notice that the vector *h[n]* is used to compute the prediction *pred[n]*. The others *h[n-...]* vectors are used as internal states to compute *h[n]*. Their size is *num_hidden*, where *num_hidden* represents the number of hidden units in the Fully Connected layer (FC) of each cell. The main problem with RNN is its incapacity to learn the connections between two cells that are far from each other [8]. This problem is known as the *Vanishing Gradient* of deep *NN*. To avoid it, Long Short Term Memory (*LSTM*) cells have been introduced by [9]. These memory cells are used in this paper; they allow an easy propagation of long term state (see vector *c* in Fig.2) along the cells with only some minor linear interactions. The *c*vector is called the cell state; it can be interpreted as the long-term state of the cells whereas the *h* vector can be interpreted as the short-term state vector. The LSTM cell is composed of 4 FC layers. In these layers, the activation function of the neurons can be Sigmoid function σ or Hyperbolic Tangent function *tanh*. These layers interact together by gates. Considering only the *g* layer is the same as having a simple RNN cell, this layer generates a Candidate vector for the cell state. The other layers are gate controllers: *f[n]* controls which part of the cell state is kept, *i[n]* controls which part of the Candidate should be added to the long term vector *c* and finally the output gate *o[n]* controls which part of the current state should go to the output *y[n]* of this time step. Once again, one can notice that *y[n]* is not the prediction *pred[n]* of the input *x[n]*, it is the output vector at time step n. Its size is *num_hidden*.The following formatting rules must be followed strictly.

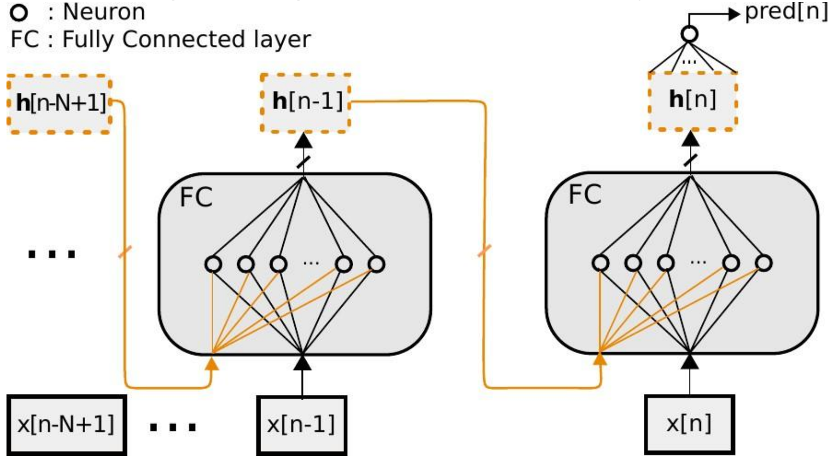


Figure 1. RNN: prediction pred[n] computed with the input sequence x of size num_step=N and the current state h[n].

3. LSTM APPLIED TO GUITAR SIGNAL EMULATION

The idea behind NN learning techniques is to minimize a distance (the Mean Square Error, MSE, is often taken for regression tasks) between a *target*called the *ground truth* and a *prediction*. In this case, the target is the output sample coming from the output of the amplifier that corresponds to an input sample *x[n]* coming from a guitar while the prediction is the sample *pred[n]* coming from the output of the emulator (i.e. the last LSTM cell of the layer for this same input sample *x[n])*. The learning process is based on a back-propagation algorithm [10] and gradient descent. The learning and emulating tasks can be divided in several steps: choose and format a data-set, describe a NN (called here a *graph*), execute the learning phase, save the model and use it for emulation. The description of these different phases is explained in this section. The Application Programming Interface (API) *Tensorflow* 1.3 [11] is used in this research for the description, the execution and the emulation of the graph. The source code can be found in [12].

3.1. Data-set

The goal is to learn the behavior of nonlinear audio systems in order to emulate them. The choice of the signal used during the learning process is thus fundamental since it has to be representative of any guitar signal.

The input signal chosen here is a guitar signal composed of two playing techniques: some single notes and some chords (a chord is composed of several notes played at the same time). The first idea was to play each note and each chord of the guitar which resulted in a very long data-set. In fact, we experimentally found out that a data-set of twenty seconds is already long enough to bring interesting results. The data-set has to be split in 3 parts: the first one is the*Training Set*, it is used in the learning phase (gradient descent and back-propagation algorithm), the second one is the *Test Set* which will be used to evaluate the model, on data than those in the training set. Finally, the third part is the *Validation Set* which is used to check that the model has not been over-fitted by selecting convenient hyper-parameters (see Section 5). The input data which is fetched to the graph must be preliminary reshaped into 3D tensor since the LSTM input needs the following shape: *[batch_size,num_step,num_feature]*. The first dimension *batch_size* is the number of input sequences *[x[n], ,x[n-num_step-1]]* that are sent at the same time to the graph in order to compute the next gradient (this is one of the hyper-parameters). The second dimension *num_step* is the length of these sequences, it corresponds to the number of LSTM cells chained in the layer. Finally,*num_feature* is the dimension of the input signal (here, *num_feature=1* since we consider the 1D vector of audio samples of the mono signal coming from the guitar).

3.2. Construction of the Graph

The construction of the graph can be divided in several steps. First, the preparation of data structure (called *placeholder*) that will contain the input and the target signals. Secondly the definition of an LSTM cell as described in Fig.2. Thirdly, the cell must be unrolled over the desired number of time steps (*num_step*). Fourthly, the output vector *y[n]* has to be sent to a simple layer of neurons to reduce its dimensionto a single sample prediction *pred[n]*. Fifthly, the MSE between all the predictions and the targets (*batch_size* predictions and targets) can be computed. Finally, the back-propagation and gradient descent can be applied.

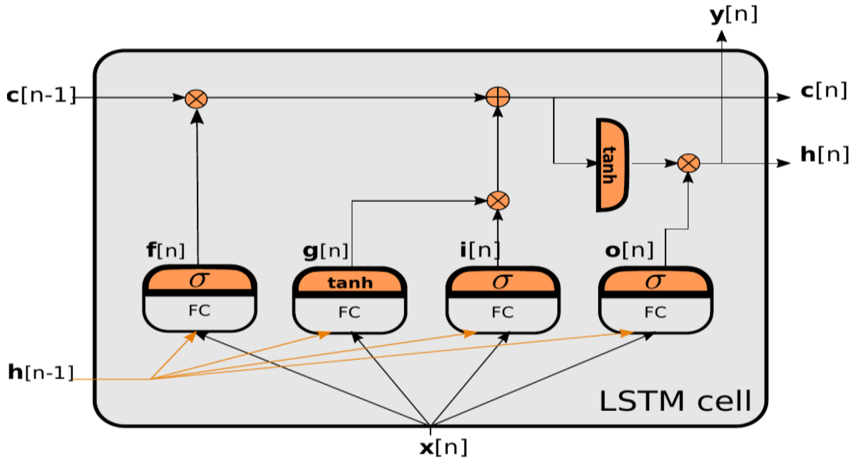


Figure 2. Long Short Term Memory cell

3.3. Execution of the Graph

The way Tensorflow works is first to place nodes on a graph where each node represents a mathematical operation. Then the graph is executed with special input nodes (called *placeholder*) containing input data from the data-set. The computation of the prediction starts and it is then possible to compute the MSE between the predictions and the targets. When all the data have been processed (called one *Epoch*), it restarts the computation until a satisfying level of accuracy is reached or until the accuracy do not evolve anymore. The graph and its parameters can then be saved in order to reuse it during the emulation phase. (More information is provided in the code example [12])

3.4. Emulation of the Graph

During the emulation phase, the graph previously saved is loaded. For real time application, the pre-processing of the guitar signal received from the sound card buffer has to be considered. Indeed, the buffer has to be reshaped in the tensor form *[batch_size,num_step,num_feature]*. The reshaping can be efficiently carried out by the *GPU* in another graph. We can use the *batch_size* parameter as the length of the input buffer coming from the sound card. Fig.3 shows how to reshape the input data. One can notice that the *feature* parameter is equal to one, so each sample *x[n]* has to be put in a list of one element. This is due to *Python* implementation where a list $a=[a_1,a_2]$ has shape=[2,] but a list $b= [[b_1],[b_2]]$ has shape=[2,1]. Note also that a vector containing the last *num_step* inputs (*last buffer*) have to be stored since the values *[x[-1]* ……. *x[-num_step]]* are needed to compute the first values of the input tensor (see Fig.3).

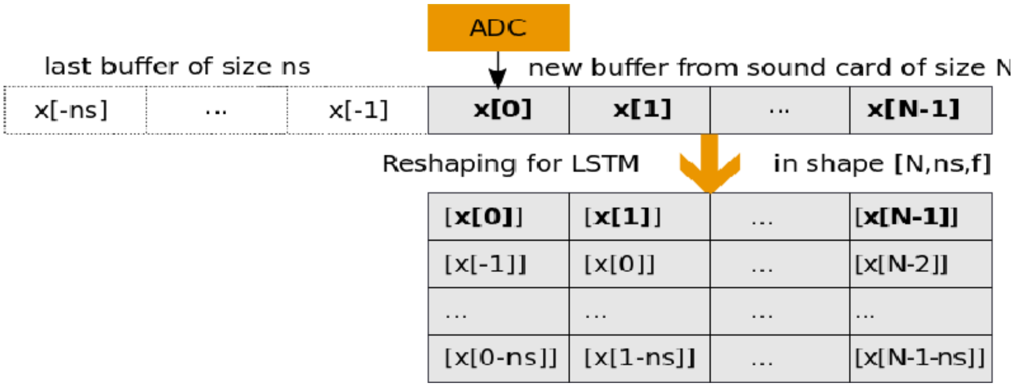


Figure 3. Reshaped input buffer of size N into LSTM input data, *ns* = *num_step*, f=feature=1

4. RESULTS

We have found that it is possible to emulate the *Engl Retro Tubes 50* at full gain (lot of distortions) with less than 1% of root mean square error RMSE between the *prediction* and the *target* (the signal that comes from the amplifier) as depicted in Fig.4 and in Fig.5 (zoom on the 800 first audio samples). The *target* signal belongs to the *validation set* and thus has never been processed by the model before. This result has been obtained with *num_step=100* and 24 hidden states. The emulation has been done by a laptop having a GPU *Nvidia gtx 1050*. As it can be seen, the curves are very close. This mean that the model is able to emulate the behaviour of the tube amplifier for a complex signal (guitar signal) that it has never seen before, in comparison with the HKISS method which can only emulate sinusoidal signals [1], this is a big improvement. The corresponding audio signals of the target and the prediction can be downloaded in *wav* format [12].

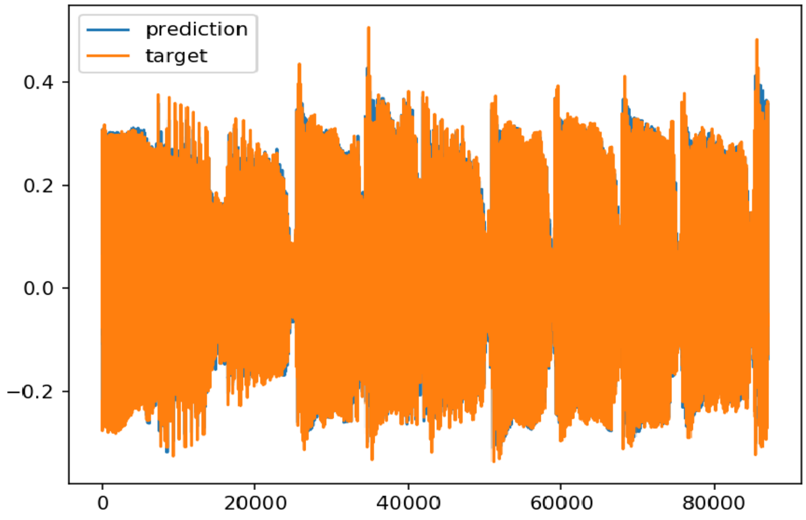


Figure 4. Temporal comparison of prediction and target signals on 2 seconds of the validation set (fs=44100Hz)

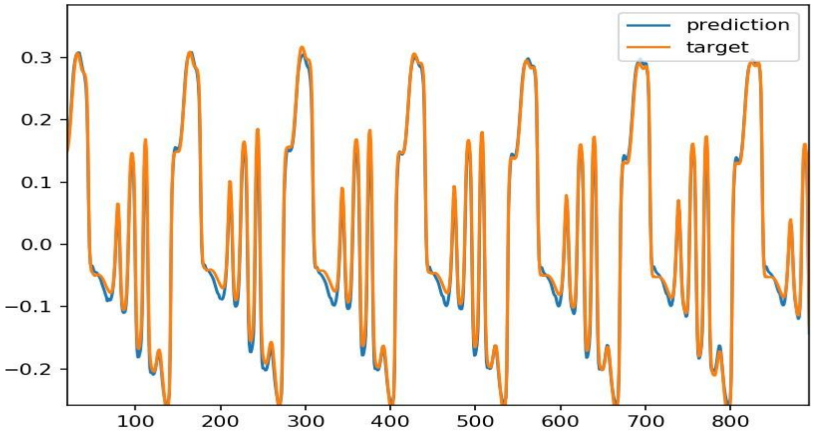


Figure 5. Temporal comparison of prediction and target signals (zoom on the first 0.02 seconds of the validation set)

4.1. Comparison with other models

There is no comparison to give with the HKISS method since this method does not support the emulation of such a complex signal as the guitar signal but a comparison with other NN structures can be made. With a Deep NN composed of 6 layers of 512 neurons (same input layer than in the LSTM case) gives a RMSE of 20% which is poor. With a *Convolutional Neural Networks [13]* structure our best result was a RMSE of 16%. The LSTM model seems thus well suited for the emulation task of a tube amplifier.

5. MODELING OF THE PARAMETERS OF THE TUBE AMPLIFIER

In the previous section an accurate model of the amplifier *ENGL Retro Tube 50* has been build. We can go further and try to include the amplifier's parameters (usually there are at least 4 parameters, the Gain parameter which sets the amount of desired distortion and 3 equalizer's parameters: Low, Middle, Treble). An interesting property of LSTM NN is that they provide an easy way to model the effects of these parameters. Indeed, the third dimension of the LSTM 's input (*num_feature*) can be used to increase the input size of data fetched to the input of the NN. For example, a two dimensional input data (*num_feature=2*) would consist of the audio sample *x[n]* and the gain *g[n]* that the amplifier had during the capture of the *target[n]*. The data-set in now composed of 3 columns *[x[n],g[n],target[n]]*.

This method with the modified LSTM NN has been applied and it also gives good results with less than 1% of RMSE. However, the model is more complex and needs more hidden units and time steps to achieve this performance. This limits the possible accuracy of the model for real time applications. Several methods have been employed in order to improve the performance of the model (i.e. smaller RMSE with smaller *num_hidden* and *num_step*): batch normalization [14], Xavier and He initialization [8], dropout [15], hyperbolic tangent and *RELU* activation function [8], faster optimizer than gradient descent (AdaGrad, RMSProp, Adam) [16]. With these methods a real-time model with less than 2% of RMSE has been found with 100 time-steps and 150 hidden units.

5.1. Hyper-parameters Exploration

LSTM have many hyper-parameters, among them are: *batch_size, num_step, num_hidden, num_layer* which are studied by letting a well-defined function to choose them randomly and train the model during a short period (ex. 3 hours). Applying this procedure many times allows the comparison of the RMSE for different sets of hyper-parameters. To speed up the learning phase, only 3 different *Gain* parameters have been taken in our training data-set. Figs.6 and 7 give the RMSE between the *target* and the *prediction* one or two layers of LSTM cells respectively. Each figure contains 2 graphs: the first one is a 3-dimensional view of the RMSE values in the (*batch_size, num_step, num_hidden*) hyper-parameters space. The second one is a projection in a *batch_size*-*num_step*plane. For real-time emulation, we would be interested to minimize the number of time steps and hidden units (lower left corner of the 2D graph). Figs.6 and 7 clearly show that the RMSE decreases if the number of hidden unit increases. Concerning the time-steps, a number between 100 and 200 seems sufficient: increasing it above this value would slow down the learning without improving the RMSE. One can also notice that the model performs slightly better with two stacked layers (a layer is composed of *num_step* chained LSTM cells). Finally, it is more difficult to have a clear opinion concerning the batch size parameter. This parameter strongly depends on the GPU used to execute the model: large value of *batch_size* allows amore accurate calculation of the gradient and takes a better advantage of the parallel abilities of the GPU. A *batch_size* value around 1000 has been found for us. In conclusion, for RT applications, choosing *[num_step,num_hidden]≈[100,150]* seems fine.

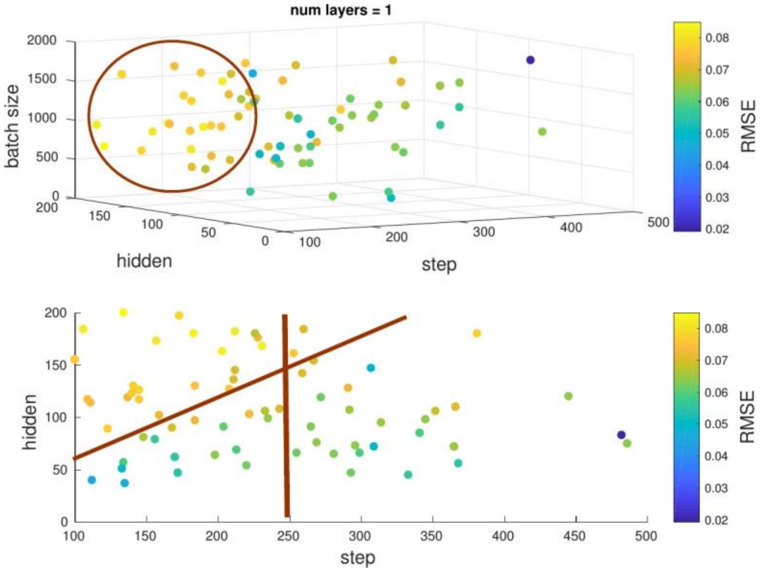


Figure 6. Comparison of RMSE between target and prediction signals using random hyper-parameters for num_layer=1

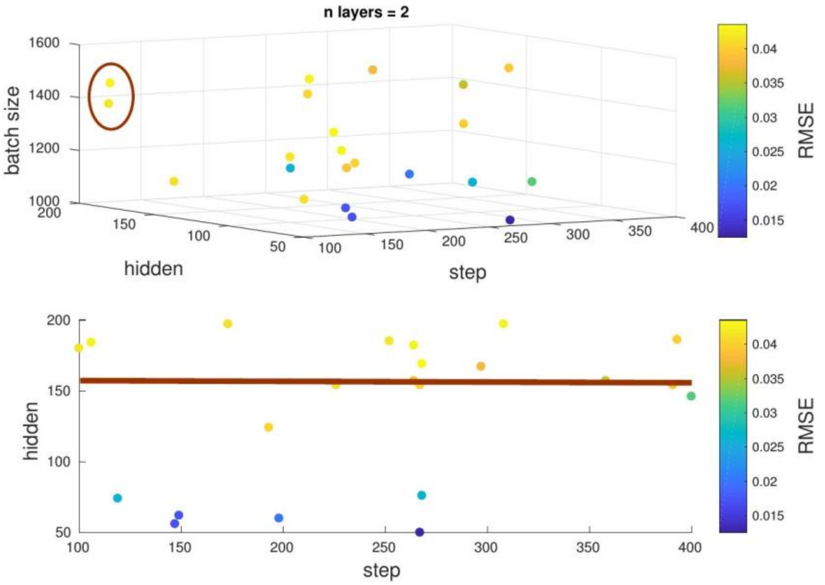


Figure 7. Comparison of RMSE between target and prediction signals using random hyper-parameters for num_layer=2.

6. CONCLUSIONS

LSTM and more generally NN have opened new perspectives to solve complex acoustic problems. The growing computational capability of new processors allows to run these models close to the real-time constraint which is important in many case such as for our emulation process. By its flexibility, the LSTM model has outperformed the cascade of Hammerstein model [1] which only was able to make accurate simulations of pure tone signals.

ACKNOWLEDGEMENTS

The Titan Xp used for this research was donated by the NVIDIA Corporation.

REFERENCES

[1] T. Schmitz & J.-J. Embrechts (2017) "Hammerstein Kernels Identification by Means of a Sine Sweep Technique Applied to Nonlinear Audio Devices Emulation", Journal of the Audio Engineering Society, Vol. 65, No. 9, pp696-710.

[2] L. Tronchin (2013) "The Emulation of Nonlinear Time-Invariant Audio Systems with Memory by Means of Volterra Series", Journal of the Audio Engineering Society, Vol. 60, No. 12, pp984-996.

[3] L. Tronchin & V.-L. Coli (2015) "Further Investigations in the Emulation of Nonlinear Systems with Volterra Series", Journal of the Audio Engineering Society, Vol. 63, No. 9, pp671-683.

[4] M. Schetzen (1980) "The Volterra & Wiener Theory of Non-linear Systems", John Wiley & Sons.

[5] T. Ogunfunmi (2007) "Adaptative Nonlinear System Identification: The Volterra and Wiener Approaches", Springer.

[6] M.Schoukens & K. Tiels (2016) "Identification of Nonlinear Block-Oriented Systems starting from Linear Approximations: A Survey", arXiv preprint arXiv:1607.01217

[7] T. Katayama & H. Ase (2018) “Linear Approximation and Identification of MIMO Wiener–Hammerstein Systems”, Automatica, Vol. 71, No. Supplement C, pp118-124.

[8] X. Glorot & Y. Bengio (2010) “Understanding the Difficulty of Training Deep Feedforward Neural Networks”, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp671-683.

[9] S. Hochreiter & S. SchmidHuber (1997) “Long Short-Term Memory”, Neural Computation, Vol. 9, No. 8, pp1735-1780.

[10] Y. Chauvin & D.-E. Rumelhart (1995) “Backpropagation: Theory, Architectures, and Applications”, Psychology Press.

[11] Tensorflow (2015) “An Open-Source Software Library for Machine Intelligence”, https://www.tensorflow.org/.

[12] T. Schmitz (2017) “LSTM Implementation for Real-Time Emulation of Nonlinear Audio System”, https://github.com/TSchmitzULG/LSTM.

[13] X. Glorot & Y. Bengio & L. Bottou & P. Haffner (1998) “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, Vol. 86, No. 11, pp2278-2324.

[14] S. Ioffe & C. Szegedy (2015) “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, International Conference on Machine Learning, pp448-456.

[15] G.-E. Hinton & N. Srivastava & A. Krizhevsky & I. Sutskever & R.-R. Salakhutdinov (2012) “Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors”, arXiv preprint arXiv:1207.0580.

[16] D. Kingma & J. Ba (2014) “Adam: a Method for Stochastic Optimization”, arXiv preprint arXiv:1412.6980.

AUTHORS

Pr. J-J. Embrechts received the degree in Electrical Engineering (1981) and the Ph.D. degree (1987) from the University of Liege (ULg). Since 1999, he is a professor at the University of Liege, in the Department of Electrical Engineering and Computer Science, where he is responsible for teaching acoustics, electroacoustics, audio and video engineering and lighting techniques. He is a member of the Board of Administration of the Belgian Acoustical Society (ABAV), a member of the Audio Engineering Society (AES), the European Acoustics Association (EAA). His current research interests are in room acoustics computer models, auralization, scattering of sound waves by surfaces, microphones and loudspeakers arrays and more generally audio signal processing.

Thomas Schmitz received the degree in Electrical Engineering (2012) from the University of Liege (ULg). His final project focused on the emulation of an electrodynamics loudspeaker including its nonlinear behavior. He is presently a Ph.D. student in Laboratory for Signal and Image Exploitation (INTELSIG) research unit of the Electrical Engineering and Computer Science (EECS) department, University of Liege, Belgium. His research interests are on signal processing, nonlinear modeling, real time emulation of guitar audio systems.

INTENTIONAL BLANK

APPLICATION TO DETERMINE OPTIMIZED PATH FOR NETWORK ROBUSTNESS

Kartikay Kaushik

Department of Electronics Engineering,
Indian Institute of Technology (Indian School of Mines), Dhanbad, India.

ABSTRACT

The recent trends of increasing unpredictability of traffic demand and the proliferation of networked devices have led to a demand for a traffic engineering application which views network robustness as a crucial factor. Robustness refers to the resilience of infrastructure networks against random and targeted failures, caused by traffic shifts, natural disasters and Denial of Service (DoS) attacks. In this paper, the author developed an application to assign the links of the network a criticality value and finds the least critical path or in other terms, the most robust path. The dynamics of the network are translated to graph metrics and the application optimizes these metrics to determine the most robust path in the network. The developed application can be further extended by incorporating its output as a feedback mechanism for another application developed for automation of the network system for robustness. The application was written in Python 2 and implemented on Ubuntu 14.04.4.

KEYWORDS

Networks, Performance prediction, Robustness, Criticality, Betweenness

1. INTRODUCTION

The study of network robustness has been an intriguing concept for scientists in the last few years. The diverse application of robustness in the fields of engineering, ecology, economics, medicine, and biology has been a motivation for scientific research across the world. With the explosion in demand for networked devices, the changes in network parameters have become highly unpredictable and thus it is important to accommodate these uncertainties while developing an application [1]. An important concept associated with a network is its maintenance. A large fraction of the cost associated with a network is spent in maintaining the network [2][3]. Hence a relatively stable system is desirable to reduce the costs involved with its maintenance. The paper aims to minimise the probability of a network failure and the costs associated with it, by finding the most stable path between two given nodesin a network.

The concept of network robustness plays an important role in improving the grade of service provided to a customer by a network service provider [4]. A low probability of network failure leads to maximal system reliability[5]. The application, when supplied with the data of flow of traffic between source and destination pairs, provides the path as an output that leads to maximum stability against external factors. The path obtained can then be used by an SDN controller to route the traffic. Hence enabling the network to withstand the changes in the parameters and adapting by changing the flow of traffic between the nodes. The output of this application

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 159–167, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80512

(primary) can serve as a feedback to another application (secondary) that is developed for network automation. The secondary application can be developed to monitor and acquire data from the network periodically and the primary application then processes the data to give the most robust path. The secondary application can later enforce the obtained path in the network thus enabling a sense of automation.

2. BACKGROUND AND RELATED WORK

There has always been a trade-off between systems that achieve high performance and systems that are robust [6]. The former aims to deliver the best performance at the cost of high sensitivity and the latter aims to provide better stability at the cost of performance. But in the recent trends, it has been observed that network and telecommunication industries opt for more robust networks [7] [8]. It is necessary to understand the Percolation Theory and the impact of node failures on the integrity of the network [9]. In this paper, the primary approach is to define the metrics that affect the network robustness.

In this paper, the concept of using graph-theoretic metrics is inspired by Dekker and Colbert [10] who used a similar approach to explain robustness. The paper [10] defines “node connectivity” as a metric to measure robustness. Their approach involved understanding the behaviour of the network in case of node failures. The papers [10] [11] concluded that node similarity and optimal connectivity are the conditions required for a network to be robust and provided methods to evaluate the same.

Freeman [12] explains the concept of “Betweenness Centrality” for node and link. The paper explains Betweenness Centrality, for node k for flow from source node i to destination node j, as the ratio of shortest paths from node i to j that pass through k. The overall Betweenness Centrality of the node k is sum of the centralities over all source-destination pairs. Link Betweenness is defined similarly.

Tezghadam and Leon-Garcian [13] define the concepts of “link criticality” and “path criticality”. The paper defined Link Criticality describes the impact of the failure of a particular link on the entire network and Path Criticality as the desirability of a path based on the criticality of the links. Also, it argues that in traffic management shortest paths need not be the best paths in all circumstances. Hence, the paper redefined the concept of Betweenness. Itstated that ifn_{ij} be the number of feasible paths between i and j and if n_{ikj} be the number of paths between i and j containing the link k. Betweenness for node k for source i and destination j is then n_{ikj}/n_{ij}. The overall betweenness of link k is the sum of the betweennesses for link k over all i and j. Hence, as Javier Mart´ınHern´andez∗ and Piet VanMieghem† stated in the paper [14], Betweenness Centrality is used as a metric to decide how critical a link in the network topology is.

3. ALGORITHM DESIGN

Tizghadam and Leon-Garcia [15] provided a theoretical basis for the design of the algorithm to calculate the metric, network criticality, to determine the robustness of the network. In this paper, the application was developed using this as a theoretical base [15].

3.1. Weight Assignment

In [15], the authors showed that if the weight increases, the goodness of that link increases. The factors increasing the goodness of the link such as available bandwidth are called as “beneficial QoS parameters” and those that decrease the goodness of the link are called the “detrimental QoS parameters” such as packet loss or length of the link. Each of these parameters is mapped to weights with an appropriate method as in [16]. In this paper, the weights are mapped by taking the product of weights of beneficial QoS parameters and dividing it with the product of detrimental QoS parameters. Thus, the weights of each link are obtained.

3.2. Network Criticality

Newman [17] argued that a probabilistic interpretation of the betweenness is defined based on random walks in a graph. A random-walk starts from a source node i, chooses a neighbor at random with equal probabilities, and gets there using the link between the source and the neighbor. The random walk continues until it reaches a specified destination *d*, where it stops. The Betweenness $b_{ck}(d)$ of a node (link) *k* for source-destination pair *s-d*is the expected number of times a random walk passes node *k* in its journey, from source *s* to destination *d*. The total Betweenness of node k is the sum of this quantity over all possible *s-d*pairs.

Consider a random walk from a source s to a destination d. The destination node is an absorbing statefor this random walk and the walk is stopped in destination.The probability of passing node *k* in next step is shown by$p_{ck}(d)$and defined as:

$$p_{ck}(d) = \left\{ \begin{array}{ll} 0 & \text{if } s = d \\ \frac{w_{ck}}{\sum_{q \in Æ(c)} w_{cq}} & \text{otherwise} \end{array} \right\} \quad (1)$$

Where A(s) is the set of adjacent nodes of *s* and $w_{ck}$is the weight of link (*s*, *k*). The first condition in equation (1) is due to the fact that the destination node *d* is an absorbing node, and any random-walk coming to this state, will be absorbed or equivalently$p_{dk}(d) = 0$.

Tizghadam and Leon-Garcia [15] thus defined node criticality for a weighted network simply as the random-walk betweenness of that node over the weight of the node.

$$n_k = \frac{b_k}{M_k} W_k = \sum_{j \in Æ(k)} w_{kj} \quad (2)$$

where $n_k b_k$, $W_k$are the criticality, betweenness, and weight of node *k* (or weighted degree of the node) respectively.$W_k$is equal to the sum of all link weights incident to node *k* (weight of link (k,j) is shown by w_{kj}).

The authors [15] derived an expression for node betweenness by making a matrix $P_d$that would describe $p_{ck}(d)$ for destination d. The probability of entering k at q^{th}step is P_d^q. The authors treated d as a fixed point and the matrices are written under this assumption. General results are obtained by adding up for all destinations.

$$B_d = [b_{ck}]_d = \left\{ \begin{array}{ll} \sum_{q=0}^{œ} P_d^q & \text{if } k \neq d \\ 0 & \text{if } k = d \end{array} \right\} = \left\{ \begin{array}{ll} (I - P_d(d|d))^{-1} & \text{if } k \neq d \\ 0 & \text{if } k = d \end{array} \right\} \quad (3)$$

Where $B_d$is the betweenness matrix for destination d. As observed, the row and column d account to 0 always and hence their removal would not affect other entities. M(i|j) denotes Matrix M without row i and column j. Hence,

$$B_d(d|d) = (I - P_d(d|d))^{-1} \quad (4)$$

If L is the Laplacian of the Matrix, in [15] the author obtained the equation for criticality by using the following equations

$L = D – W$ where $D = \text{diag}(W_1, W_2, W_3, W_4 \ldots W_n)$and W is the weight matrix (5)

$P_d(d|d) = D(d|d)^{-1} \times W(d|d)$

$I - P_d(d|d) = I - D(d|d)^{-1} \times W(d|d)$
$I - P_d(d|d) = D(d|d)^{-1} \times L(d|d)$ (6)

$B_d(d|d) = L(d|d)^{-1} \times D(d|d)$ (7)

The reduced inverse of the Laplacian matrix is given as

$$L(d|d)^{-1}_{ck} = l^+_{ck} - l^+_{cd} - l^+_{dk} + l^+_{dd} \quad (8)$$

Where l^+_{ck} is the entry of row s and column k of the Moore-Penrose inverse of L.
$(B_d(d|d))_{ck} = (l^+_{ck} - l^+_{cd} - l^+_{dk} + l^+_{dd}) \times W_k$

$\frac{[b_{ck}]_d}{M_k} = l^+_{ck} - l^+_{cd} - l^+_{dk} + l^+_{dd}$

For total betweenness of node k, the effect of all source-destination pairs is considered.

$$\frac{b_k}{M_k} = \frac{1}{M_k}\sum_c \sum_d [b_{ck}]_d = \frac{1}{M_k}\sum_c \sum_d \frac{[b_{ck}]_d + [b_{dk}]_c}{2}$$

$$\frac{b_k}{W_k} = \sum_c \sum_d \frac{l^+_{dd} - l^+_{cd} - l^+_{dc} + l^+_{cc}}{2}$$

$$\frac{b_k}{M_k} = \sum_c \sum_d \frac{S^+_{dd} - 2S^+_{cd} + S^+_{cc}}{2} = \frac{1}{2}\sum_c \sum_d T_{cd} = \frac{1}{2}T \quad (9)$$

$T = \sum_c \sum_d T_{cd} = \sum_c \sum_d (l^+_{cc} + l^+_{dd} - 2\,l^+_{cd})$

Hence, $b_k = W_k \frac{T}{2}$ for a node k or $b_{ij} = w_{ij}T$ for a link (i,j) (10)

T is known as the network criticality. Less the network criticality, lesser the sensitivity to changes. Hence more stable.

3.3. Optimization of Network Criticality

According to Tizghadam and Leon-Garcia [15], the equation for an optimal weight set W*, and using the concept of optimisation from [18] [19]

$$C\frac{\&T}{\&w_{ij}} + T \geq 0 \ \forall\, (i,j) \in E \quad (11)$$

$$\text{And}\ \frac{\&T}{\&w_{ij}} = -2n\, ||L^+_i - L^-_j||^2 \quad (12)$$

Substituting (10) and (12) in (11)

$$\frac{\&b_{ij}}{\&w_{ij}} = w_{ij}\,[-2n\, ||L^+_i - L^-_j||^2\,] + T \quad (13)$$

By substituting C = -1 in (11) and substituting (13)

$$\frac{\&b_{ij}}{\&w_{ij}} = w_{ij} \left[2n \,||L_i^{+} - L_j^{-}||^{2}\,\right] \tag{14}$$

Equation (14) gives the value of optimised cost of each link.

3.4. Finding Most Robust Path

The cost of each link is determined from Equation (14). These costs are assigned as the new weights of the links. The least cost path can be found by using several techniques like Dijkstra's Algorithm [20]. This least cost path is also the most robust path [21].

3. APPLICATION DEVELOPMENT

The application was written in Python 2 in Ubuntu 14.04.4. The input of the application is taken in the form of an adjacency matrix wherein the elements of the matrix are the weights of the links. The Laplacian is obtained by Equation (5). The term D is obtained by constructing a diagonal matrix from the matrix obtained by taking the sum along each row. Following the equation (9), a matrix is made for each s-d pair. The sum of the elements of the matrix gives the network criticality.

From, Equation (10), the product of Network Criticality with weight matrix results in a betweenness matrix that gives betweenness for every source and destination pair. Equation (14) results in a cost matrix that gives the cost for every source and destination pair. The application takes source and destination node as an input from the user. By using Dijkstra's Algorithm, the least cost path is obtained between the given source and destination node. Hence, the most robust path is obtained between the given nodes.

4. EVALUATION

In this section, experiments are conducted on different network topologies and run the application for different source and destination nodes. The first line of input is the size of the network or the number of nodes present in the network. From the second line, the adjacency matrix of the graph is given as an input. The application gives the network criticality, betweenness matrix and cost matrix for the given graph. When supplied with the initial point and the output point, the output is the most robust path between the nodes.

4.1. Topology - 1

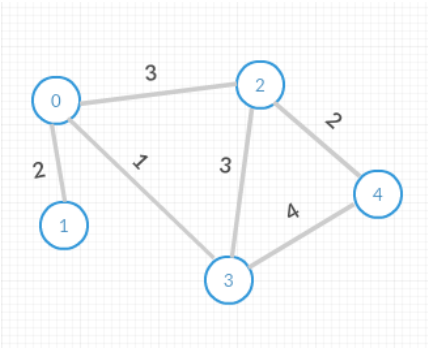
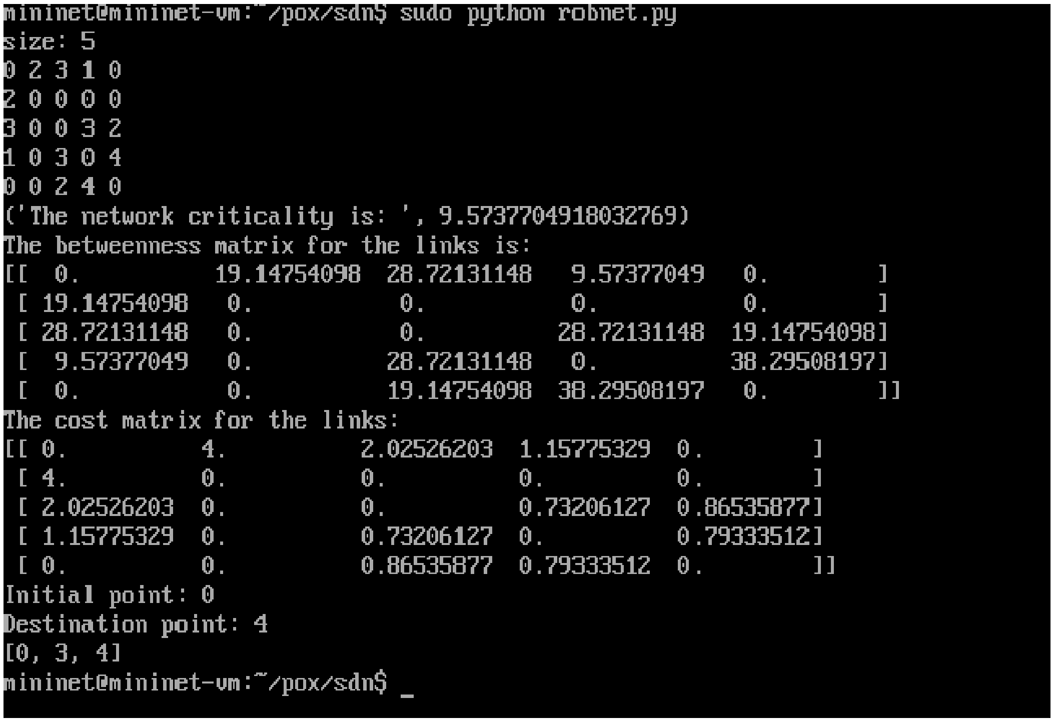


Figure 1. Topology – 1

The output for Figure 1 for initial point as 0 and destination point as 4 is given in Figure 2. The size of the matrix is 5 and the adjacency matrix is obtained from Figure 1.



```
mininet@mininet-vm:~/pox/sdn$ sudo python robnet.py
size: 5
0 2 3 1 0
2 0 0 0 0
3 0 0 3 2
1 0 3 0 4
0 0 2 4 0
('The network criticality is: ', 9.5737704918032769)
The betweenness matrix for the links is:
[[  0.           19.14754098  28.72131148   9.57377049   0.        ]
 [ 19.14754098   0.           0.           0.           0.        ]
 [ 28.72131148   0.           0.          28.72131148  19.14754098]
 [  9.57377049   0.          28.72131148   0.          38.29508197]
 [  0.           0.          19.14754098  38.29508197   0.        ]]
The cost matrix for the links:
[[ 0.          4.          2.02526203  1.15775329  0.        ]
 [ 4.          0.          0.          0.          0.        ]
 [ 2.02526203  0.          0.          0.73206127  0.86535877]
 [ 1.15775329  0.          0.73206127  0.          0.79333512]
 [ 0.          0.          0.86535877  0.79333512  0.        ]]
Initial point: 0
Destination point: 4
[0, 3, 4]
mininet@mininet-vm:~/pox/sdn$ _
```

Figure 2. Output for Topology – 1

Thus, from the above application, it has been observed that the path from 0 to 3 to 4 is the most stable path between 0 and 4.

4.1. Topology - 2

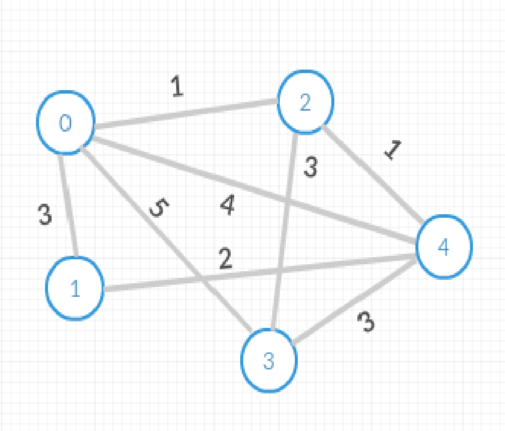
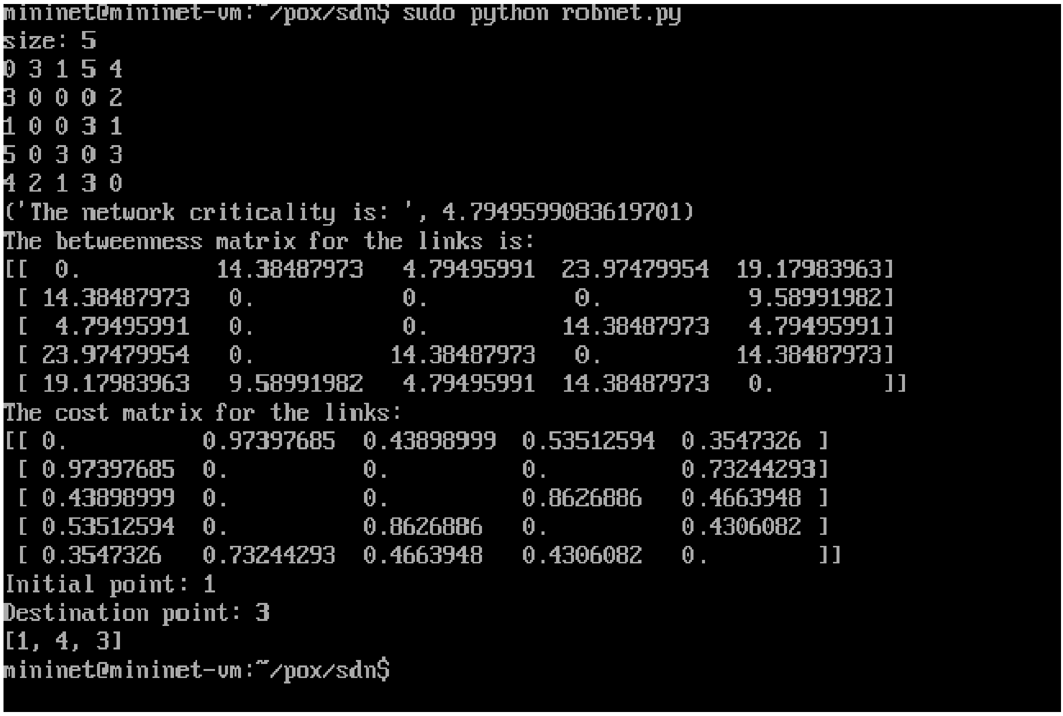


Figure 3. Topology – 2

The output for Figure 3 for initial point as 1 and destination point as 3 is given in Figure 4. The size of the matrix is 5 and the adjacency matrix is obtained from Figure 3.



```
mininet@mininet-vm:~/pox/sdn$ sudo python robnet.py
size: 5
0 3 1 5 4
3 0 0 0 2
1 0 0 3 1
5 0 3 0 3
4 2 1 3 0
('The network criticality is: ', 4.7949599083619701)
The betweenness matrix for the links is:
[[  0.          14.38487973   4.79495991  23.97479954  19.17983963]
 [ 14.38487973   0.           0.           0.           9.58991982]
 [  4.79495991   0.           0.          14.38487973   4.79495991]
 [ 23.97479954   0.          14.38487973   0.          14.38487973]
 [ 19.17983963   9.58991982   4.79495991  14.38487973   0.        ]]
The cost matrix for the links:
[[ 0.          0.97397685  0.43898999  0.53512594  0.3547326 ]
 [ 0.97397685  0.          0.          0.          0.73244293]
 [ 0.43898999  0.          0.          0.8626886   0.4663948 ]
 [ 0.53512594  0.          0.8626886   0.          0.4306082 ]
 [ 0.3547326   0.73244293  0.4663948   0.4306082   0.        ]]
Initial point: 1
Destination point: 3
[1, 4, 3]
mininet@mininet-vm:~/pox/sdn$
```

Figure 4. Output for Topology – 2

Thus, from the above application, it has been observed that the path from 1 to 4 to 3 is the most stable path between 1 and 3.

3. CONCLUSION AND FUTURE WORK

In this paper, an application was developed to determine an optimized path for network robustness. The developed application has been tested for different network topologies. It has been observed that the path obtained in the output had the least cost and hence most stable.

The work can be further extended towards automation of networks by dividing an application into three parts. They are data acquisition, data processing and policy enforcement. The data acquisition and policy enforcement stages involve interacting with the physical layer to receive data and implement policies. The application developed in this paper can be used in the data processing stage. The data acquired from the traffic is converted into graph-theoretic metrics. The data is processed and the path obtained from it is implemented. A continuous and periodic monitoring of the traffic and the updating of policies induce a sense of automation in the network. By reducing the human intervention, the quality of service can be highly increased. It can solve the problem of costs involved with the maintenance and the delay in response.

ACKNOWLEDGEMENTS

This research was supported in part by Department of Electronics Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad and Centre for Development of Advanced Computing, Pune.

REFERENCES

[1] David G. Messerschmitt, What the NII could be: A User Perspective, The Unpredictable Certainity

[2] E. Arcaute, R. Johari and S. Mannor, Network Formation: Bilateral Contracting and Myopic Dynamics.

[3] Amir Ranjbar, Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) Foundation Learning Guide, Ch 3.

[4] Gerald R. Ash, Robust Design of Dynamic Routing Networks, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Volume 5, 1991.

[5] Ioannis P. Chochliouros and George A. Heliotis, Optical Access Networks and Advanced Photonics: Technologies and Deployment Strategies, Page 37

[6] D. Applegate and E. Cohen. Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: understanding fundamental tradeoffs. In SIGCOMM, pages 313–324, 2003.

[7] R. Boutaba, W. Szeto, and Y. Iraqi. DORA: Efficient Routing for MPLS Traffic Engineering. Journal of Network and Systems Management, 10(3):309–325, September 2002.

[8] K. Kar, M. Kodialam, and T. V. Lakshman. Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications. IEEE Journal on Selected Areas in Communications, 18(12):2566–2579, Dec. 2000.

[9] D. Stauffer and A. Aharony. Introduction to Percolation Theory.Taylor and Francis.London, 1994.

[10] A. H. Dekker and B. D. Colbert.Network Robustness and Graph Topology.Australasian Computer Science Conference, 26:359–368, Jan. 2004.

[11] Ali Tizghadm, Alireza Bigdeli, Alberto Leon-Gracia and Hassan Naser, Joint Optimization resources and routes for minimum resistance from communication networks and power grids, Springer 2012.

[12] L. C. Freeman. Centrality in Networks: I. Conceptual Clarification. Social Networks, (1):215–39, 1978/79

[13] A. Tizghadam and A. Leon-Garcia.A Robust Routing Plan to Optimize Throughput in Core Networks. ITC20, Elsvier, pages 117–128, 2007.

[14] Classification of graph metrics Javier Mart´ın Hern´andez∗ and Piet VanMieghem†, November, 2011.

[15] A. Tizghadam and A. Leon-Garcia.Autonomic Traffic Engineering for Network Robustness.Vol 28, No. 1, Jan. 2010

[16] P. Van Mieghem and F. A. Kuipers. Concepts of Exact QoS Routing Algorithms .IEEE/ACM Transactions on Networking, 12(5):851–864, October 2004.

[17] M. Newman. A Measure of Betweenness Centrality Based on Random Walks. ArXiv cond-mat/0309045., 2003.

[18] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. Convex Analysis and Optimization. Athena Scientific, April 2003.

[19] S. Boyd and L. Vandenberghe. Convex Optimization.Cambridge University Press, 2004.

[20] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 2nd Edition.

[21] David Hock, Matthias Hartmann, Christian Schwartz, and Michael Menth, Effectiveness of Link Cost Optimization for IP Rerouting and IP Fast Reroute

AUTHOR

The author is pursuing B.Tech (Honours) degree in Electronics and Communication Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad and will be graduating in May 2018. His field of interest includes SDN, VLSI and computer architecture. His research experience includes an internship in DRDO and CDAC, Pune.

INTENTIONAL BLANK

A SILENT HACK DETECTION BASED ON DEEP-LEARNING TECHNIQUE

Nuha Almozaini, Yasmin Alateeq, Noura Alrajeh and Saleh Albahli

IT Department, College of Computers, Qassim University, KSA

ABSTRACT

Sharing information has been democratized with the rise of social networks. Consequently, increasing the usage of Social Network, especially Twitter platform, leads to growing malicious activities. With a silent hack, a hacker can continuously dig around to control over victim's account. In this paper, an observed direct impact to users' security and privacy has been identified. Therefore, we address hidden tactics in the problem specific feature engineering with detailed results to show how deep leaning classifiers are promising direction to understand sentiment than classical machine learning. Thus, we focus on the state of the art Deep learning techniques by constructing a model to detect behavioral changes of users. Our evaluation shows that working with just classical machine algorithms to analyse social data do not achieve higher performance than deep learning algorithms. This will open directions for using deep learning for similar problems. Moreover, our results demonstrate the shortages of classical Machine Learning classifiers compared to Deep learning and how they can be mitigated.

KEYWORDS

Deep learning, Machine learning, silent hacking, social data, behaviors, analysis, Twitter.

1. INTRODUCTION

With the rise of social networks, some people exploit them for bad behaviors. Therefore, it becomes a huge risk to young people's ethics because of their limited capacity for self regulation and susceptibility to peer pressure. With millions of tweets every day, these lead to growing malicious activities.

Technology now is more than collecting and preparing information to users and organizations. Technology seeks insight and knowledge, and that is what we look for in this paper. With Machine Learning behind the scenes, the field that concerns with how machine can learn from experiences, handling the intelligence part to go deeply in data and predict what's coming is our way of the paper.

Activities of users by flooding the Internet with data and sharing content in social networks specifically as a lifestyle must have implicit things in their activities. These are interesting, significant and most importantly abnormal which is a goal to either governments or organizations that target users. These activities that users act online are commonly called behaviors.

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 169–176, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80513

Therefore, this paper aims to find such behaviors that threaten privacy and ethics in communities, countries and people by collecting enough data, setting it up and then finally subjecting it to analysis stages involving machine learning algorithms and predictions. Then, it will end up with a data product that captures what would risk people all around the world.

Thus, we hypothesize that, on one side, working with just classical machine algorithms to analyse social data do not achieve higher performance than deep learning algorithms. As such, this paper attempts to show how deep leaning classifiers are promising direction to understand sentiment than classical machine learning. Besides, we study the effects of a behavioral change of users by using Python with trained different models to expose behaviors and get subjectivity of micro-blogging content. Accordingly, we attempt in this paper to find out more about attitudes, relationships and connectivity between users of Twitter by taking the powerful of deep learning techniques.

Problem Statement: Social network penetration worldwide is ever-increasing. The increased worldwide usage of Online Social Network (OSN) that leads to growing malicious activities. Twitter is one of the most social networks have infected accounts. Therefore, silent hacking on Twitter try to reach active users resulting in negatively impacting other genuine Twitter users. Thus, we aim to focus on the nature of social networking and how users among social media react and behave. We also aim to detect whoever acts abnormal and penetrates user's privacy putting both acts in deeper analysis stages to bring insights from these behaviors. Ultimately, we attempt to protect normal users to avoid damaging their image.

Research Questions: Our research questions focus on two main theme:

- What are the main challenges when analyzing behavior in Social Networks and Twitter interactions?

- How can these challenges be addressed by using state-of-the-art machine learning technologies and algorithms?

Proposed Approach: People are concerned with the security of their accounts and their private information in the accounts and they might not know if they were hacked because some attackers don't leave a trace, so we are going to seek this trace.

Our solution is to capture behaviors that meet our expectations of social media behaviors and lead to reasoned view of how people interact over the internet. Practically, our solution focus on the state of the art deep learning classifiers by constructing a model to detect behavioral changes of users. We label collected tweets as spam or non-spam and adverts or not. In addition, identifies four features (URL, Hashtag, Media, sensitive information) that lead us inferencing them as normal or abnormal. Then, used mechanism to analyze each collected tweet manually and independently to show how Deep Learning classifiers achieve better performance than classical Machine learning classifiers.

Contributions: The primary contributions of our paper may be summarized as follows:

- To construct a model to detect behavioral changes of users in online social media.

- To show how Deep Learning classifiers has an advantage over classical Machine learning classifiers.

Organization: Section 2 provide the related work, followed by the discussion of the benefit of using Semantic Analysis over Sentiment Analysis. Section 3 gives an overview of data collection. The candidate features to identify hacked tweets are presented in section 4. The experimental design, evaluation and results are shown in section 5. Finally, we conclude the paper in section 6 with an outlook on future work.

2. RELATED WORK

Since millions of users on Twitter tweet constantly in their informal languages, the issue with this type of analysis is analyzing words and how to deal with informal phrases. It may contain acronyms, abbreviations, slang words, misspelled words, non-opinion words, sarcasm, etc. Hence, the challenge is to detect neutral words that will help removing non-opinion words [1]. In addition, Twitter often keeps sensitive information about users like their locations secret. Therefore, privacy issue plays a major role of collecting dataset for this type of information [2].

Sentiment Analysis has been used for long time to classify words based on linguistic artifacts that show sentiment and syntax patterns to link subject with the sentiment classification. On the other hand, a more reliable approach is called semantic analysis which help to cluster different data rely on similarity instead of current classification such as positive/negative/neutral [3][4][5].

Savage et al.[6] use graph based analysis to observe suspicious behaviors in Online Social Networks (OSN) and categorized them into three different types: inappropriate content share, silent hacking, and fake promotional accounts. For the first type, they use it to catch accounts/nodes that have very high in-degree as well as very less number of friends; users with such behavior are usually hiding themselves using fake identities. For the second type, since hackers will not be visible and obvious, an active node gets a pattern different from usual which it increases visibly but the out-degree remains the same and therefore it is called “silent”. The third and final type is handled by seeking out-degrees for a node. Hence, if most of the out-degrees targeting one node, this act will be categorized to the fake promotional accounts since this type of accounts intentionally promote specific node/user.

Bravo-Marquez et al. [7] propose word-level classification to show how to generate opinion lexicons from unlabelled tweets. They use Sentiment Analysis to classify words either positive, negative or neutral. Tweets are represented by using two vector: bag of words and a semantic vector based on word-clusters. Finally, they show that the clusterbased vectors are better than the bag of words vectors.

El Kassiri et al. [8] [9] show that semantic similarity measure and RDF graphs achieve high performance for link prediction. They propose an Ontology called ActOnto to share common activities made by communities in social networks. However, as mentioned before, there are other frameworks used to analyses opinions of users in social networks by using the sentiment analysis.

3. DATA COLLECTION

Dataset is always an asset and an indicative part of data science experiments. In the proposed work, dataset prepared continuously and carefully throughout the experiment. Furthermore, a survey was made for 380 twitter users to clarify the kind of actions to be observed from raw data. The main impact of the dataset is sampling (Extracting subset of a dataset), and sampling has been done regularly. When a sample is acquired, labelling manually is applied to the sample, then preparation and preprocessing steps are implemented to make the sample conforms to the experiment requirements. Last step is training the classifiers then evaluate them for prediction and if the prediction accuracy is low, alternatives are taken i.e. obtain more data, redo the experiment and so on.

The experiment is implemented with 3576 tweets, 1694 are labelled abnormal and 1882 are normal.

Data Analysis: In our experiment, we analyze data based on tweet contents. The data were trained using different Machine Learning classifiers (Table 5.1) to be able to compare accuracy of the classifiers regarding the problem domain. 10% of the data was excluded from the training phase to be used to test the models and 90% was in the training phase.

4. CANDIDATE FEATURES

Based on the domain knowledge, we identified hacked tweets based on six candidate features: URL, Spam, Ads, Media, Hashtag and Sensitive information [10][11][12][13].

URL: hackers target users by posting links of malicious websites in their tweets. Since links lead victims to the sites, they often use this method in addition to URL shortened services which can hide the targeted URL. So, hackers tend to use such services to post links to their websites. This feature identified by true/false whether a URL in a tweet or not.

Spam: spam accounts on Twitter post same content many times and maybe have a small changing of the tweet. Accordingly, they target to send same content to many users.

Ads: it was also noticed that advertisement content usually reflects some malicious content characteristics so, advertisement content has been selected as the third feature. An example of Ads tweet can be shown as follows:

```
Click to #win #Hellraiser: The Scarlet Box on Blu-ray with @HeyUGuys
          https://t.co/afsq2qGB1Q https://t.co/hMq5wZl3G2
```

Media: identified whether media contents are included or not. This feature can be indicated a spam and is like sub-feature of spam feature, therefore media has been selected as the forth feature.

Hashtag: infected accounts try to attract legitimate users to read their tweets by posting multiple unrelated tweets using trending hashtags. These accounts hope to reach more viewers quickly, so they use trending and popular hashtags in their tweets. Therefore, this feature is candidate for detecting spammers too.

Sensitive information: Attackers normally lead users to their malicious contents, so they take advantage of the basis of human behavior. Thus, users are led by their instincts which sometimes overcome morals and ethics. Moreover, the leverage of adult content can be a trigger for human instincts that leads them to explore such content, therefore we take sensitive content as a candidate feature.

After picking appropriate features, tweets have been cleaned feasibly in a way that does not take out the treasure of the problem-related anomaly data [14]. Cleansing process accomplished using Python scripts and luckily the dataset has only English tweets so the cleansing process was for eliminating emojis, symbols as well as irrelevant URLs.

When data was collected directly from Twitter feed, there were URLs in tweets that were not practically activities of users sharing external content. These URLs were "Quote Tweet" activities. A quote Tweet is very normal activity in Twitter platform where users comment on others' tweets but as new tweets published directly to their own feed so that their followers can see what they commented on, But the tweets come with a URL of the tweets being commented-on. Now Twitter replaced these URLs with small boxes containing the commented-on tweets for readability but not at data collection. That is why these URLs were excluded programmatically from the dataset.

5. EXPERIMENTS

This section discusses our experimental evaluation with different popular machine learning models, including Deep Learning, Support Vector Machine (SVM), Random Forest and Naive Bayes as summarized in Table 5.1. The same parameter initialization is utilized when comparing multiple optimization classifiers.

Dataset and Feature Selection:

Acquiring data is an easy task since datasets are publically available online, but acquiring an appropriate and suitable data for specific domain can be very difficult task so data scientists may end up with data generation tools to obtain relevant and appropriate data. Since the proposed work is a classification problem, there was no dataset available online that meets the need of this work and ready for classifying abnormal behaviors. Thus, raw data has been collected from a freely available dataset. The collected data is a subset of the dataset, which is Tweets attribute, also known as feature. Feature selection is a task that requires deeper look into the problem to get the most useful and relevant features to the problem domain. However, for constructing a model to handle text data, the following preprocessing techniques were applied to increase text mining accuracy. First technique is Tokenization which takes lines of text then turns them into individual separated words. The second is Stop Words Elimination, and stop words are the unimportant words that do not add any meaning for text analysis experiments, like the word "an". The third is Stemming, and this technique turns words to their stem like the words "Closed" and "Closing" both will be "Close" after Stemming. The fourth and last technique is Transform Cases which turns uppercase to lowercase so that the "Read" and "read" words will be considered the same after this process.

The dataset has been enhanced several times using features engineering. The models have been improved using text processing techniques such as tokenization to end up with a better accuracy as shown in Table 5.1.

Table 5.1: Overview of experimental results

Deep Learning	**95.57%**
Random Forest	94.90%
Support Vector Machine (SVM)	93.67%
Naïve Bayes	91.72%

Experimental Results and Discussion for classical algorithms: For classical Machine Learning classifiers, Table 5.1. shows that SVM resulted robust classification capabilities that even with major and minor changes of the experiment it stays at high accuracy, near 90% and ended up with 93.67%. **Random Forest**, on the other hand, has shown sensitivity regarding text preprocessing and feature engineering. It shows very low accuracy at the beginning of the experiment when there are four features, tweets, URL, spam and ads and only tokenization for text preprocessing. But it surprisingly increased to 94.90% after optimizing text preprocessing methods and feature engineering. For **Naive Bayes**, it shows good enough result from the beginning and was very flexible at optimizing. It kept increasing while improving the experiment, until it ended up with 91.72% accuracy.

Experimental Results for Deep Learning: In Table 5.1, the Deep learning classifier resulted 95.57% of the tweet correctly, which shows the best performance among other classical ML algorithms. However, Deep Learning did not work well at the beginning, so after enhancing many feature engineering and Neural Network, it is resulted much higher than other compared classifiers. The overall number of layers used is six; two for input/output and four hidden layers. Dropout was applied to all hidden layers with value of 0.2 and dropout is Deep Learning approach to avoid overfitting so that the model can be tested on unseen data. The Adam optimization algorithm [15] used throughout. The ReLU activation [16] used for our experiment for all layers except the output layer which used Sigmoid activation. So, after all these optimizations for layers' architecture, Deep Learning classifier is reached 95.57% accuracy.

6. CONCLUSIONS AND FUTURE WORK

Increasing the usage of Online Social Network (OSN) leads to growing malicious activities. Twitter is one of the most social networks have infected accounts. Our paper aims to show that Deep Learning classifiers has gained advantage over classical Machine learning classifiers. As such, we study the effects of a behavioral change of users by using Python with trained different models to expose behaviors and get subjectivity of microblogging content. Accordingly, we show in this paper how to find out more about attitudes, relationships and connectivity between users of Twitter by taking the powerful of deep learning techniques. In detail, four algorithms were applied to the problem and what was remarkable is that deep learning went beyond expectations for the prepared small portion of data which agree with our hypothesis.

Since we aim at to detect behavioral changes of users in online social network, our further research plans to extend the proposed work to build a semantic-based model using different ontology techniques.

REFERENCES

[1] F. Atefeh and W. Khreich, “A survey of techniques for event detection in Twitter,” Computational Intelligence, vol. 31, no. 1, pp. 133–164, 2015.

[2] J. Schmidhuber, “Deep Learning in neural networks: An overview,” Neural Networks, vol. 61. pp. 85–117, 2015.

[3] A.R. Guess, “Sentiment Analysis v. Semantic Analysis: A much more statistically reliable approach is semantic analysis.” [Online]. Available: http://www.dataversity.net/sentiment-analysis-v-semantic-analysis/. [Accessed:18-Jan-2018].

[4] H. Saif, Y. He, and H. Alani, “Semantic sentiment analysis of twitter,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, vol. 7649 LNCS, no.7 PART 1, pp. 508–524.

[5] R. Giovanetti and L. Lancieri, “Model of computer architecture for online social networks flexible data analysis: The case of Twitter data,” in 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016, pp. 677–684.

[6] D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang, “Anomaly detection in online social networks,” Social Networks, vol. 39, no. 1, pp. 62–70, 2014.

[7] F. Bravo-Marquez, E. Frank, and B. Pfahringer, “From Unlabelled Tweets to Twitter-specific Opinion Words,” Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR’15, pp. 743–746, 2015.

[8] A. El Kassiri and F. Z. Belouadha, “ActOnto: An extension of the SIOC standard for social media analysis and interoperability,” in Colloquium in Information Science and Technology, CIST, 2015, vol. 2015–Janua, no. January, pp. 62–67.

[9] A. El Kassiri, F. B.-I. S. T. and, and undefined 2015, “Towards a unified semantic model for online social networks analysis and interoperability,” in 10th International Conference on Intelligent Systems: Theories and Applications (SITA), 2015, pp. 1–6.

[10] X. Ruan, Z. Wu, H. Wang, and S. Jajodia, “Profiling Online Social Behaviors for Compromised Account Detection,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 176–187, 2016.

[11] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, “Towards Detecting Compromised Accounts on Social Networks,” IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 4, pp. 447–460, 2017.

[12] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, “Detecting spammers on social networks,” Neurocomputing, vol. 159, no. 1, pp. 27–34, 2015.

[13] S. Rosenthal, N. Farra, and P. Nakov, “SemEval-2017 task 4: Sentiment analysis in Twitter,” Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 502–518, 2017.

[14] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553. pp. 436–444, 2015.

[15] D. P. Kingma and J. L. Ba, “Adam: a Method for Stochastic Optimization,” International Conference on Learning Representations 2015, pp. 1–15, 2015.

[16] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” Proceedings of the 27th International Conference on Machine Learning, no. 3, pp. 807–814, 2010.

TRAFFIC SIGN CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK

Nemanja Veličković[1], Zeljko Stojković[2] , Goran Dimić[2], Jelena Vasiljević[2]
and Dhinaharan Nagamalai[3]

[1]University Union, School of Computing,
Knez Mihailova 6/VI, 11000 Belgrade, Serbia
[2]Institute Mihajlo Pupin, University of Belgrade,
Volgina 15, 11 000 Belgrade, Serbia
[3]Wireill, Australia

ABSTRACT

Artificial Neural Networks enables solving many problems in which classical computing is not up to task. Neural Networks and Deep Learning currently provide the best solutions to problems in image recognition, speech recognition and natural language processing. In this paper a Neural Network, more specific - Convolutional Neural Network solution for the purpose of recognizing and classifying road traffic signs is proposed. Such solution could be used in autonomous vehicle production, and also similar solutions could easily be implemented in any other application that requires image object recognition.

KEYWORDS

Artificial neural network, convolutional neural network, classification, traffic sign

1. INTRODUCTION

Computer Vision Science (CVS) as a part of Artificial Intelligence (AI) deals with fetching, processing, analysing and understanding images. Main goal of the CVS is to simulate human vision through utilizing AI algorithms, which is still far from possible.

Classical computing for long tried solving these problems by modelling problem specific algorithms. Toady Deep Learning (DL) techniques provide us solutions in which a computer can "learn" the solution algorithm without us having to explicitly model the problem. The main trade off being we need a set of labelled data on which the computer will pick up the statistical differences which will enable it to discriminate the objects into classes.

One of the major problems that AI tries to solve is categorizing the input data based on the previous inputs. An example of that is e-mail spam filtering or classifying e-mails into *spam* and *not spam* categories. There are several ways of training such models, one of them being *supervised learning* where the idea is to use pairs of input and output data to predict the output data as close as possible, and the other *unsupervised learning* where only input data is used, and the model is trying to spot statistical differences in the input data.

This document describes the use of Neural Networks (NN), more specific - Convolutional Neural Networks (CNN) in classifying traffic road signs. CNN was chosen as a method of solving this

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 177–186, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80514

problem because is their great success in solving similar problems. In order to get some background information in NN and CNN field chapters 2 and 3 will go through some basic theory of NN and CNN, how they work, how they are trained and their basic components. After that in chapter 4 we will show a way of using CNN for solving the problem of classifying the traffic road signs.

2. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network is a set of connected neurons in such a way that it loosely mimics human brain. The outputs of one layer are connected as the inputs of the next layer of neurons. An example of such network can be seen in Figure 1. A typical ANN is composed of 3 basic layers: *input layer*, *hidden layer* and *output layer.*

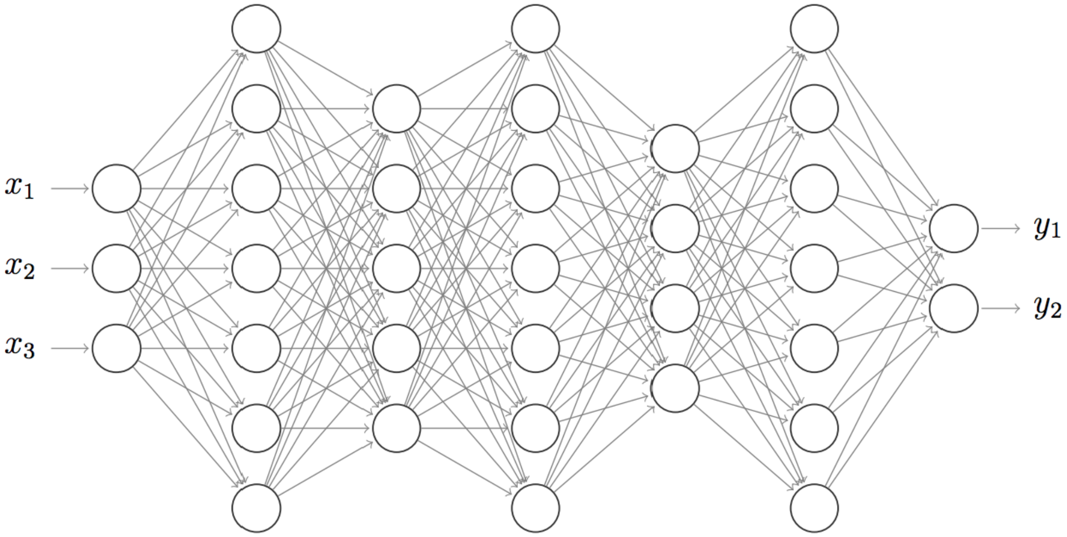


Figure 1. Set of connected neurons forming an ANN

Such network would get its inputs as a set of values [x_1, x_2, x_3] so we can consider the neurons labelled with x as the input layer of the network. Neurons of the input layer are connected to the neurons of the first layer of the hidden layer. The hidden layer neurons are connected to other neurons and their outputs are not visible as a network output (hence the term hidden). The hidden layer neurons are presented as unlabelled neurons in Figure 1. The outputs of the last layer ofthe hidden layer neurons are connected to the neurons of the output layer. The set [y_1, y_2] makes the output values of the network.

2.1. Neuron

An Artificial Neuron is a mathematical function conceived as a model of biological neurons. Artificial neurons are elementary units of an ANN. The neuron receives one or more inputs and sums them to produce an output (or activation). An example of such a neuron can be found in Figure 2.

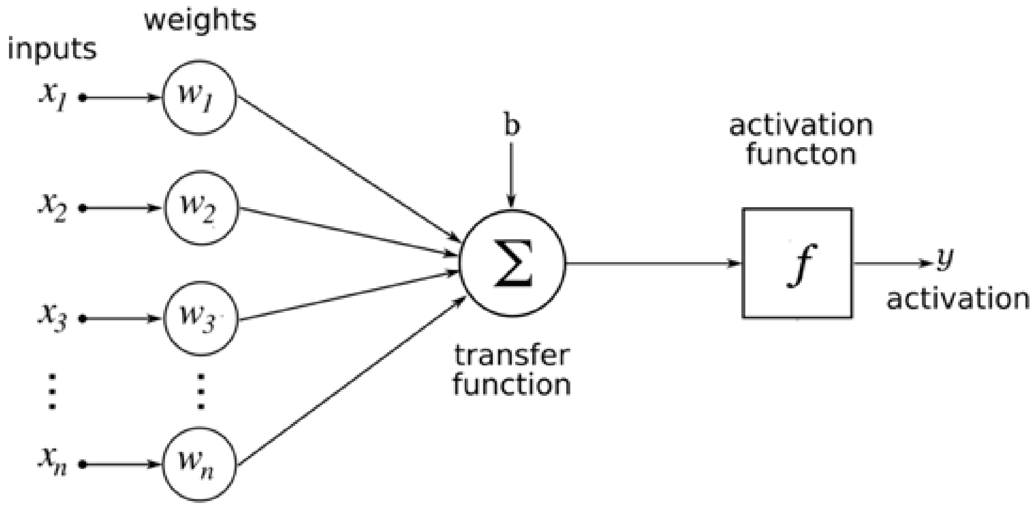


Figure 2. Neuron with inputs [x_1,..,x_n] and an output y

Mathematically the neuron from the Figure 2 is presented by the following expression:

$$y=f\left(\sum_{i=1}^{n}\square x_i\cdot w_i+b\right)$$

The expression holds two unknown values and an unknown function. Values [w_1,..,w_n] represent the weights of the corresponding inputs, value b is the bias and the function *f* is the activation function. The input values are first weighed, then they are put through the transfer function (in our case the transfer function is SUM - ©) ant lastly the output of the transfer function is passed through the activation function *f* which will be activated dependent of the value of the transfer function and the activation threshold to produce the output of the neuron. Depending of the application the transfer function must not only be the SUM function and functions like MAX, MIN, AVG, etc.

2.2. Activation function

Neuron without an activation function (or with an activation function $f(x) = x$) represents an ordinary linear combination of input parameters. Activation function must be nonlinear in order for the NN to learn nonlinear functions. Simply speaking outputs of the hidden layer neurons with linear activation function would all be linear combination of the input parameters. With nonlinear activation function neural network has far more possibilities over a simple perceptron network (a network only built with simple nodes only containing an input and an output).

There are many types of activation functions, two most commonly being used are:

- TANH – mapping $f:(-,)(-1,1)$:

$$f(x)=\frac{e^{2x}-1}{e^{2x}+1}$$

and

- SIGMOID – mapping $f:(-,)(0,1)$, which could be used when we are expecting only positive outputs:

$$f(x)=\frac{1}{1+e^{-x}}$$

2.3. Training an Artificial Neural Network

The majority of practical machine learning uses supervised learning. Supervised learning is where you have input values and corresponding output values (input data has been labelled) and you use this data to learn the mapping function of the input to the output. The goal is to approximate the mapping function so well that when you introduce new input data you can predict the output values for that data. It is called supervised learning because the process of training the ANN can be taught as a teacher supervising the learning process. We know the correct answers, the ANN will repeatedly predict the answer, the error will be back propagated, and the ANN will know a bit more about the data. The learning stops when the ANN achieves an acceptable prediction success rate.

Unsupervised learning is where you only have input data and no corresponding output values. The goal for unsupervised learning is to model the underlying structure in the data in order to learn more about the data. It is called unsupervised learning because there is no corresponding output data and the error cannot be calculated and back propagated through the ANN. Unsupervised learning is commonly used to solve clustering problems.

2.3.1. Back propagation of errors

Back propagation is a form of supervised learning. It composes of two fazes *forward pass* and *backward pass*. A collection of weights makes the ANN model because it is an attempt to model data’s relationship to training data labels. Model normally starts out bad and ends up less bad, changing over time as the ANN updates its parameters. This is because the ANN is born in ignorance, it does not know which weights and biases will translate the input data into desired result. So, it starts guessing. The first set of input is given to the ANN. It makes a prediction the best it can. The weights map the input to a set of guesses.

$$guess = input * weights$$

The ANN will then compare the predicted result with the actual truth value. The difference between the truth value and the predicted value is the ANN prediction error.

$$error = truth - guess$$

The network measures that error and walks the error back over its model, adjusting weights to the extent that they contributed to the error.

$$adjustment = error * weight$$

The ANN will then add the adjustment value to the corresponding weight, and the next time the ANN makes a prediction it will be that much closer to the truth result. This way we can fine tune the ANN to give better and better scores after each iteration. This algorithm is repeated until we are satisfied with the way the ANN predicts the result.

3. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks are deep Artificial Neural Networks used primarily to classify images (name what they see), cluster them by similarity (photo search), and perform recognition with scenes. CNN are the algorithms that can identify faces, individuals, traffic signs, tumours and many other aspects of visual data. The efficiency of CNNs in image recognition is one of the main reasons why the world has woken up to the efficiency of deep learning algorithms.

CNN works with multidimensional data $x \in R^n \times R^n$ as opposed to regular ANN which works with scalar data. CNN ingest data as tensors, and tensors are matrices of numbers with additional dimensions. Neurons of the CNN convolution layers are called *feature maps* and their weights are called *kernels*.

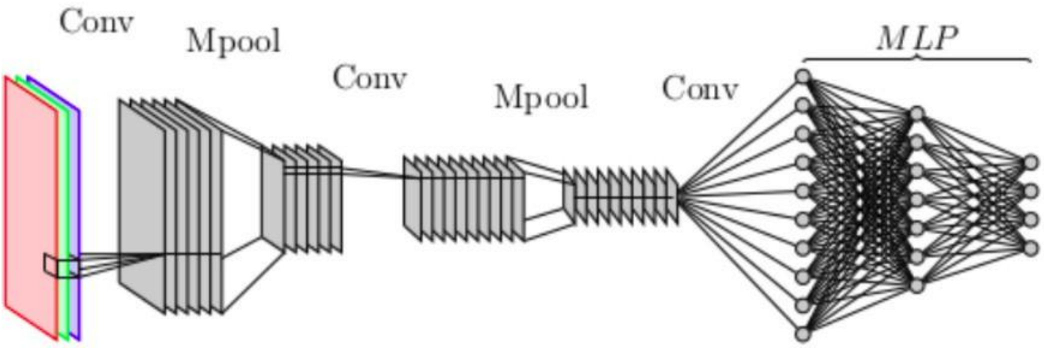


Figure 3. An architectural example of a Convolutional Neural Network

An example of a basic CNN structure can be seen in Figure 3. The input of such a network can either be a 3-channel coloured image or a one channel monochromatic image. The inputs width and height are the same of the image width and height and the colour channels dictate the depth of the input data. If the image is 300x300 pixels in width and height and images colours are written in RGB format and we input each colour as a single channel, then the dimensions of input data would be 300x300x3. Since monochromatic images only have one colour the dimensions of such input data would be 300x300x1. After the input layer come alternately connected *convolution layers* and *pooling layers.* After the convolutional and pooling layers come the fully-connected layers much like a regular NN and it is responsible to compute the score. In this way CNN transforms the original image layer by layer from the original pixel values to the final score. There are two types of layers found in CNN:

- Convolutional layer
- Pooling layer

3.1. Convolutional layer

Convolutional layer is the core building block of a CNN that does most of the computational work. The parameters of a convolutional layer consist of a set of learnable filters (*kernels*). Every filter is small spatially but extends the full depth of the input volume. An example of such filter would be a filter with dimensions 5x5x3 (5 pixels for width and height and depth 3 because the image has 3 channels RGB). During the forward pass we convolve (slide) each filter across the width and height of the input data and compute dot products between the entries of the filter and the input at any position. The result of this convolution will be a 2-dimensional activation map that gives the responses of that filter at every spatial position. The CNN will learn the filters that activate when they see some type of visual feature. Now that we have a set of all the filters from all of the layers we have the CNN model on which we can test our results. An example of a convolutional layer can be seen in Figure 4.

3.1.2. Local connectivity

When dealing with high-dimensional inputs such as images it is impractical to connect a neuron with every neuron from the next layer. Instead we connect each neuron to only a local group of the input volume. The spatial extent of this is a parameter called the *receptive field* of the neuron. The extent of connectivity along the depth is always equal to the depth of the input data.

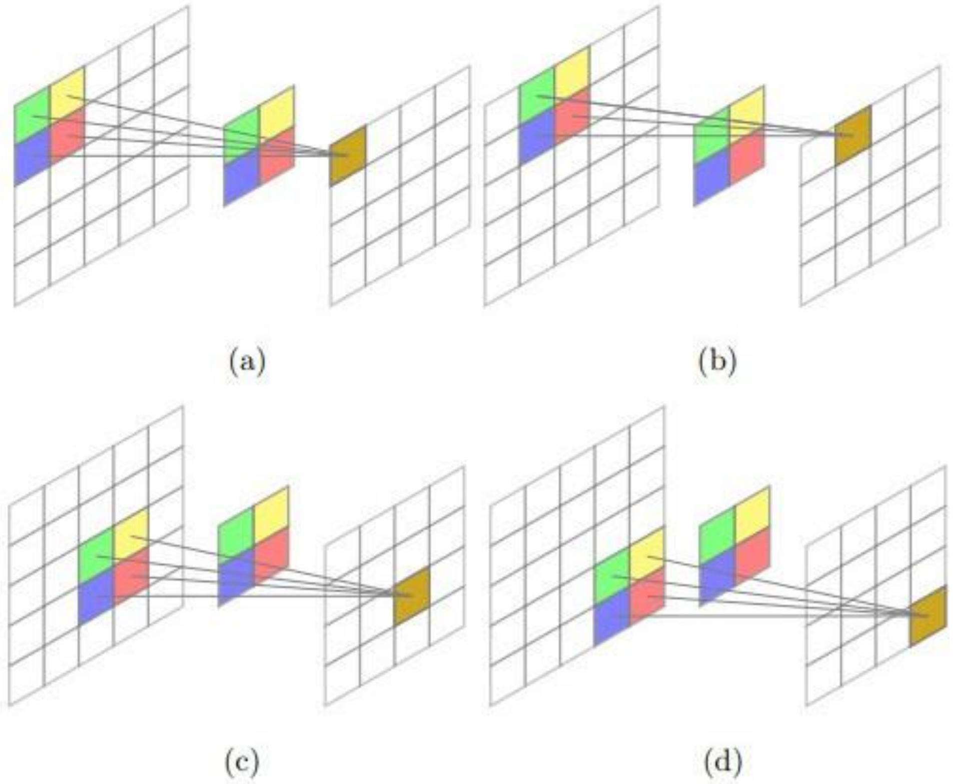


Figure 4. Graphical convolution example

3.2. Pooling layer

Pooling layers have a function of reducing the size of the of the filter outputs and are usually periodically placed between the convolutional layers of the Convolutional Neural Network. The task of reducing the number of outputs is needed in order to reduce the number of parameters and calculations in the network, and also to prevent *over fitting* the network. The pooling layer operates independently on every depth slice of the input and resizes it spatially. The most common form is a pooling layer with filters of size 2x2, applied with a stride of 2 down samples every depth slice in the input by 2 along both width and height, discarding 75% of activators. The depth dimension remains unchanged.

There are several types of pooling functions to choose from, some of them being:

- MAX pooling
- MIN pooling
- AVG pooling
- L2-norm pooling

An example of pooling layer can be seen in Figure 5.

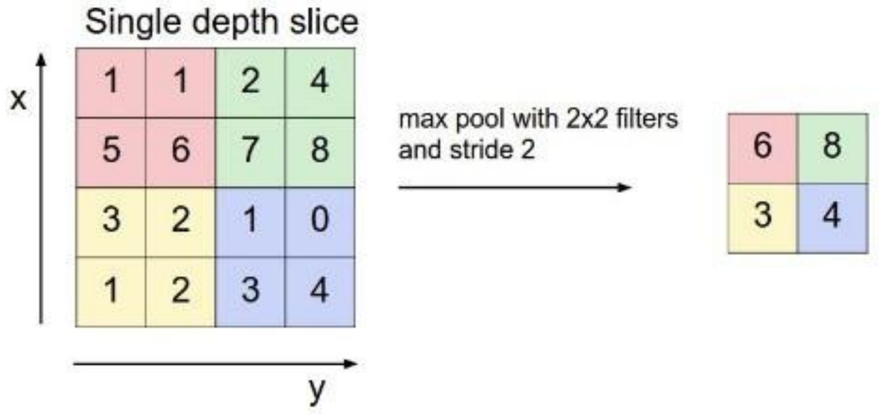


Figure 5. Graphical pooling layer example using max pool filter with

When using MAX pooling function, it is common to keep an index of the max activation, so the back propagation could be done more efficiently. Also, since during the forward propagation only the max values are taken into account the back propagation will update only those max values when error correcting. Many people dislike the pooling operation and think they can go without it. Common ways of getting rid of pooling layer is to have convolutional layers with larger stride so the convolution would in effect reduce the size of the output dramatically.

3.3. Convolutional Neural Network parameters

Convolutional Neural Network parameters (also called hyperparameters) are a set of variables used to control the network. It is a must that the hyperparameters are initializes before training the network. Hyperparameter optimization is a process of finding close-to-optimal CNN parameters. Common hyperparameters in CN networks are:

- **Learning rate** is one of the most important parameters of a neural network. ANN convergence is dependent on this parameter. Optimizing these parameters is possible by using random values and training the network and afterwards choosing the best parameter value from the lot

- **Epoch number** parameter can easily be optimized using early stop technique, simply put you would print the error after each epoch you can choose how long you will train you network

- **CNN architecture** presents a set of decisions which model our network. Choosing the right number and orientations in the network, choosing the input data dimensions, kernel dimensions, strides, etc… It is needed to construct the network in such a way that the network is large enough to solve our particular problem, but not too large so we get a sub-performing network

- **Activation function** is in most cases the same value across the ANN. In CNN applications most commonly used activation function is ReLU (Rectifier Linear Units)

- **Weights initialization** should be done very carefully in order not to slow down the network in the beginning stages of training process. Commonly used weight initialization function is *uniform distribution*

- **Cost function** is an important ANN parameter on which the convergence of the ANN is dependent on. It is used to calculate the error between the output and the input data. Most commonly used cost function is *mean squared error.*

- **Strides** represents the value by which the filter is going to move along the width or height of the input data

- **Padding** represents adding an additional frame of empty pixels to the input data, most common of value 0. Padding is used to ensure the ANN does not neglect the outer pixels

4. TRAFFIC SIGNS CLASSIFICATION USING CNN

Recognising traffic sign is a huge step in producing reliable autonomous vehicles. This chapter will show you how we implemented ANN based solution to classifying traffic signs. Since we already covered the basic theory in previous chapter the assumption is that the reader knows the basics about ANN.

The dataset consists of 43 different traffic sign classes and each image being of size 30x30 pixels totalling about 1200 images in total. The dataset contained an equal amount of images of each class. All of the data from the dataset was normalized to [0,1] values. All of the images are RGB coloured meaning their full dimension is 30x30x3. The dataset was split into two sets:

- **Training set** consisting of about 80% of the dataset, and

- **Test set** consisting of the rest 20% of the dataset

4.1. The architecture

After trying several different CNN architecture examples we decided to go with the following architecture based on the LeNet.

- Input with dimensions 30x30x3

- First CONVOLUTION layer with 32 output filters, kernel size 3x3, stride 1x1, non-biased

- First MAXPOOL layer with kernel size 2x2

- Second CONVOLUTION layer with 64 output filters, kernel size 5x5, stride 1x1, non-biased

- Second MAXPOOL layer with kernel size 2x2
- First FULLY CONNECTED layer with 1600 inputs and 1200 outputs
- Second FULLY CONNECTED layer with 1200 inputs and 1024 outputs
- Third FULLY CONNECTED layer with 1024 inputs and 512 outputs
- Final OUTPUT layer with 512 inputs and 43 outputs (one output per class)

All layers are without padding and all layers having activation function ReLU. The network weights were initialized by using uniform distribution and the selected loss function was mean squared error. The implementation was performed in deeplearning4j framework library.

4.2. Results and discussion

We trained and evaluated the CNN several times and we averaged accuracy of about 80%. While there is still room for improvement we certainly showed that traffic sign classification using CNN and DL is possible. We noticed that the CNN had problems classifying similar shaped signs. For instance, the speed limit traffic signs. Both of them are circular signs and only differing in the value written in the centre of the sign. This can be caused by many factors like quality of the images, similar (if not the) same sign shapes. This could be fixed by adding new Convolutional Layers that could enable the network to learn even more.

5. CONCLUSION AND FUTURE WORK

The application of neural networks in traffic sign recognition has recently been a field of study. With this paper we showed that using Convolutional Neural Network in traffic sign classification is possible. This application, with some improvements, could be used in autonomous vehicle production and in traffic safety.

Future work would be to increase the prediction success rate of the Convolutional Neural Network described in the previous chapter. This could be achieved by using larger sets of data on which the CNN would learn. This means preparing more traffic sign images and feeding them to the CNN. Also, the performance of the CNN could be increased by using more layers, but this would mean the complexity of the network is dramatically increased and with that the time for the network to learn is also increased. Also, some type of object localization algorithms could be used with this CNN in order for the network to be capable of working with video stream.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education and Science, Republic of Serbia, Science and Technological Development grant III 43002.

REFERENCES

[1] W.A. Awad and S.M. ELseuofi, “Machine Learning Methods for Spam e-mail Classification”, International Journal of Computer Science & Information Technology, 2011

[2] V.R.Kulkarni, Shaheen Mujawar1 and Sulabha Apte, “Hash Function Implementation Using Artificial Neural Network”, International Journal on Soft Computing, 2010

[3] Santaji Ghorpade, Jayshree Ghorpade and Shamla Mantri, “Pattern Recognition Using Neural Networks”, International Journal of Computer Science & Information Technology, 2010

[4] Andrej Karpathy, Convolutional Neural Networks for Visual Recognition, Stanford University, 2016

[5] Amritpal Kaur, Madhavi Arora, “Neural network based Numerical digits Recognization using NNT in Matlab”, International Journal of Computer Science, 2013

[6] Matija Folnović, “Deep neural architectures for object localization”, Sveučilište u Zagrebu, 2015

[7] Aravindh Mahendran, Andrea Vedaldi, “Understanding Deep Image Representations by Inverting Them”, Computer Vision Foundation, 2015

[8] https://www.coursera.org/specializations/deep-learning

[9] https://deeplearning4j.org/

[10] http://benchmark.ini.rub.de/?section=gtsrb&subsection=news

AUTHORS

Nemanja Veličković is currently pursuing a M.Tech. degree in Computer Science and Engineering at Računarski fakultet, University Union, School of Computing, Belgrade, Serbia and has completed a B.E in Computer Science and Engineering also at Računarski fakultet. Currently working as a full stack developer at Ticketmaster.

AUTHOR INDEX