

David C. Wyld
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

9th International Conference on Advanced Information Technologies
and Applications (ICAITA 2020)
July 11~12, 2020, Toronto, Canada



AIRCC Publishing Corporation

Volume Editors

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

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

ISSN: 2231 - 5403

ISBN: 978-1-925953-23-7

DOI: 10.5121/csit.2020.100901- 10.5121/csit.2020.100921

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

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

Preface

The 9th International Conference on Advanced Information Technologies and Applications (ICAITA 2020) July 11~12, 2020, Toronto, Canada, 9th International Conference on Data Mining & Knowledge Management Process (CDKP 2020), 9th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2020), 6th International Conference on Networks and Communications (NCO 2020), 6th International Conference on Control, Modeling and Computing (CMC 2020), 6th International Conference on Software Engineering (SOFT 2020), International Conference on Machine Learning & Trends (MLT 2020), International Conference on Advanced Natural Language Processing (AdNLP 2020) was collocated with 9th International Conference on Advanced Information Technologies and Applications (ICAITA 2020). 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 ICAITA 2020, CDKP 2020, SAI 2020, NCO 2020, CMC 2020, SOFT 2020, MLT 2020 and AdNLP 2020 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, ICAITA 2020, CDKP 2020, SAI 2020, NCO 2020, CMC 2020, SOFT 2020, MLT 2020 and AdNLP 2020 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 ICAITA 2020, CDKP 2020, SAI 2020, NCO 2020, CMC 2020, SOFT 2020, MLT 2020 and AdNLP 2020.

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
Dhinaharan Nagamalai (Eds)

General Chair

David C. Wyld,
Dhinaharan Nagamalai,

Organization

Southeastern Louisiana University, USA
Wireilla Net Solutions, Australia

Program Committee Members

A. Alsultanny,
Abdulhamit Subasi,
Ahmed A. Elngar,
Akhil Gupta,
Alia Karim Abdulhassan,
Aman Jatain,
Ana Luísa Varani Leal,
Archit Yajnik,
Assia DJENOUHAT,
Benyettou Mohamed,
Bouhorma Mohammed,
BRAHAMI Menaouer,
Callistus Ireneous Nakpih,
Carol Anne Hargreaves,
Chien-Cheng Yu,
Christophe Gueyux,
Denivaldo Lopes,
Diab Abuaiadah,
Dinh-Thuan Do,
Dipti Kapoor Sarmah,
Djamel Eddine,
Ee-Peng Lim,
El Mostapha Aboulhamid,
Emad Awada,
Emilio Jiménez,
Ernesto C. Marujo,
Feras N. Hasoon,
Ghanshyam Prajapati,
Govardhan Hegde K,
Hala Abukhalaf,
Hazem El-Gendy,
Hiromi Ban,
Hu, Yu-Chen,
I.Thirunavukkarasu,
I-Ching Hsu,
Islam Atef,
Iyad Alazzam,
Jackson Akpojaro,
Joao Gama,
Johannes K. Chiang,
John Tass,
Jose Alfredo F. Costa,
Jyotir Moy Chatterjee,

University of Baghdad, Iraq
Effat University, Saudi Arabia
Beni-Suef University, Egypt
Lovely Professional University, India
University of Technology, Iraq
Amity University, India
University of Macau, China
Sikkim Manipal University, India
University Badji Mokhtar Annaba, Algeria
University of Sciences and technology, Algeria
Faculté des sciences et techniques Tangier, Morocco
National Polytechnic School of Oran, Algeria
St. John Bosco's College of Education, Ghana
National University of Singapore, Singapore
Hsiuping University of Science and Technology, Taiwan
Université de Bourgogne Franche-Comté, France
Federal University of Maranhao - UFMA, Brazil
Waikato institute of technology, New Zealand
Eastern International University, Vietnam
Symbiosis International (Deemed University), India
University of Biskra, Algeria
Singapore Management University, Singapore
Universite de Montreal, Canada
Applied Science University, Jordan
University of La Rioja, Spain
Professor of Operations Research at ITA, Brazil
Sohar University, Sultanate of Oman
Gujarat Technological University, India
Manipal University, India
Palestine Polytechnic University, Palestine
Ahrm Canadian University, Egypt
Fukui University of Technology, Japan
Providence University, Taiwan
Manipal University, India
National Formosa University, Taiwan
Alexandria university, Egypt
Yarmouk University, Jordon
University of Africa, Nigeria
University of Porto, Portugal
National Chengchi University, Taiwan
University of Patras, Greece
Federal University, Brazil
Lord Buddha Education Foundation, Nepal

Kavita Singh,	Yeshwantrao Chavan College of Engineering, India
Kin-Choong Yow,	University of Regina, Canada
Klenilmar L.Dias,	Federal Institute of Amapa-Macapa Campus, Brazil
Koczy T.Laszlo,	Budapest University of Technology, Hungary
Mahmood Hashemi,	Beijing University of Technology, China
Maissa HAMOUDA,	SETIT & ISITCom, University of Sousse, Tunisia
Manyok Chol David,	University of Juba, South Sudan
María Hallo,	Escuela Politécnica Nacional, Ecuador
Maryam AL-Jabri,	Sohar University, Sultanate of Oman
Marzan Tasnim Oyshi,	Daffodil International University, Bangladesh
Masaru Kitsuregawa,	Tokyo University, Japan
Mohamed Issa,	Zagazig University, Egypt
Mohamed Sahbi Bellamine,	Carthage University, Tunisia
Mohammad A. M. Abushariah,	The University of Jordan, Jordan
Mohammad Ashraf Ottom,	Yarmouk University, Jordan
Mohammad Hajjar,	Lebanese University, Lebanon
Mohammad Khaled Al_Maghasbeh,	Ministry of education, Jordan
Mohammed GH. AL Zamil,	Yarmouk University, Jordan
Mueen Uddin,	Effat University Jeddah, Saudi Arabia
Muthukumar Murugesan,	Karpagam University, India
Nadia Abd-alsabour,	Cairo University, Egypt
Nawaf Omar N. Alsrehin,	Yarmouk University, Jordan
Nguyen Dinh Lau,	University of Danang, Vietnam
Omar Al-harbi,	Jazan University University, Saudi Arabia
Piotr Malak,	University of Wroclaw, Poland
Radhika K R,	B M S College of Engineering, India
Rao Kotagiri,	University of Melbourne, Australia
Rohan De Silva,	Central Queensland University, Australia
Ruaa Alsabah,	University of Kerbala, Iraq
Sa'adah Hassan,	Universiti Putra Malaysia, Malaysia
Said Agoujil,	University of Moulay Ismail Meknes, Morocco
Saleh Al-Daaajeh,	Abu Dhabi polytechnic, UAE
Sébastien Combéfis,	ECAM Brussels Engineering School, Belgium
Shahid Ali,	AGI Education Ltd, New Zealand
Shirish Patil,	Independent/Industry, USA
Surekha Kamath,	Manipal University, India
Susmita Gupta,	Indian Institute of Technology, India
Utku Kose,	Suleyman Demirel University, Turkey
Venkata Duvvuri,	Purdue University, USA
veslava osinska,	Nicolaus Copernicus University, Poland
Vilas M. Thakare,	SGB Amravati University, India
Wajid Hassan,	Indiana State University, USA
Xiao Zhang ,	University of Denver, USA
Xiaodong Liu,	Edinburgh Napier University, UK
Xiaofei Zhang,	Nanjing University, China
Yahya Almayali,	University College of Humanity Studies, Iraq
Yang Li,	Beihang University, Beijing, China
Youssef Tahe,	Center of guidance and planning, Morocco
Zoran Bojkovic,	University of Belgrade, Serbia
Zuhal Tanrikulu,	Bogazici University, Turkey

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)



Artificial Intelligence Community (AIC)



Soft Computing Community (SCC)



Digital Signal & Image Processing Community (DSIPC)



Organized By



Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

9th International Conference on Advanced Information Technologies and Applications (ICAITA 2020)

Comparison of GNSS Patch Versus GPS L1 Patch Antenna Performance Characteristic.....01 - 12
Gholam Aghashirin, Hoda S. Abdel-Aty-Zohdy, Mohamed A. Zohdy, Darrell Schmidt and Adam Timmons

Classification of Fatigue in Consumer-grade EEG Using Entropies as Features.....13 – 26
Muhammad Azam, Derek Jacoby and Yvonne Coady

Project of Sorting System for Plastic Garbage in Sorting Plant Based on Artificial Intelligence.....27 – 35
Janusz Bobulski and Mariusz Kubanek

9th International Conference on Data Mining & Knowledge Management Process (CDKP 2020)

Evaluating and Validating Cluster Results.....37 - 47
Anupriya Vysala and Joseph Gomes

Data-Driven Techniques for Music Genre Recognition.....49 - 63
Sergio Santiago Rentería, Jesus Leopoldo Llano and Francisco Javier Cantú-Ortiz

Prediction of Cancer Microarray and DNA Methylation Data using Non-negative Matrix Factorization.....65 - 81
Parth Patel, Kalpdrum Passi and Chakresh Kumar Jain

A Framework for Capturing and Analyzing Unstructured and Semi-Structured Data for a Knowledge Management System.....83 - 94
Gerald Onwujekwe, Kweku-Muata Osei-Bryson and Nnatubemugo Ngwum

A Systematic Review of Natural Language Processing for Knowledge Management in Healthcare.....275 - 285
Ganga Prasad Basyal, Bhaskar P Rimal and David Zeng

9th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2020)

Facial Expression Recognition Using combined Pre-Trained Convnets..... 95 - 106
Raid Saabni and Alon Schclar

Follow then Forage Exploration: Improving Asynchronous Advantage Actor Critic 107 - 118
James B. Holliday and T.H. Ngan Le

Automated Classification of Banana Leaf Diseases using an Optimized Capsule Network Model 119 - 130
Bolanle F. Oladejo and Oladejo Olajide Ademola

An Adaptive Utilization of Convolutional Matrix Methods on Sliced Hippocampal Neuron Cell Segmentation with an Application Interface 131 - 139
Neeraj Rattehalli and Ishan Jain

6th International Conference on Networks and Communications (NCO 2020)

Wireless Sensor Networks Simulators and Testbeds 141 - 159
Souhila Silmi, Zouina Doukha, Rebiha Kemcha and Samira Moussaoui

An Enhanced Lucene based System for Efficient Document/Information Retrieval 161 - 167
Alaidine Ben Ayed, Ismail Biskri and Jean-Guy Meunier

Multiple Layers of Fuzzy Logic to Quantify Vulnerabilities in IoT..... 169 - 187
Mohammad Shojaeshafiei, Letha Etzkorn and Michael Anderson

6th International Conference on Control, Modeling and Computing (CMC 2020)

A New Intelligent Power Factor Corrector for Converter Applications 189 - 199
Hussain Attia

A Flow Simulation in the Foaming Process 201 - 211
Karel Fraňa, Jörg Stiller and Iva Nová

6th International Conference on Software Engineering (SOFT 2020)

DOCPRO: A Framework for Building Document Processing Systems 213 - 224
Ming-Jen Huang, Chun-Fang Huang, Chiching Wei

International Conference on Machine Learning & Trends (MLT 2020)

Object Detection in Traffic Scenarios - A Comparison of Traditional and Deep Learning Approaches 225 - 237
Gopi Krishna Erabati, Nuno Gonçalves and Hélder Araújo

**International Conference on Advanced
Natural Language Processing (AdNLP 2020)**

Using Holographically Compressed Embeddings in Question Answering.... 239 - 250
Salvador E. Barbosa

**A Semantic Question Answering in a Restricted Smart Factory
Domain Attaching to Various Data Sources.....** 251 - 274
Orçun Oruç

COMPARISON OF GNSS PATCH VERSUS GPS L1 PATCH ANTENNA PERFORMANCE CHARACTERISTIC

Gholam Aghashirin¹, Hoda S. Abdel-Aty-Zohdy¹, Mohamed A. Zohdy¹,
Darrell Schmidt² and Adam Timmons³

¹Department of Electrical and Computer Engineering,
Oakland University, Rochester, Michigan, USA

²Department of Mathematics and Statistics,
Oakland University, Rochester, USA

³Department of Mechanical Engineering, McMaster University,
Hamilton, Canada

ABSTRACT

Antenna module is a vital component of automated driving systems, it should function as needed in dGPS, HD map correction services, and radio and navigation systems. The proposed antenna model for GPS only patch antenna operating at 1.57542 GHz and the GNSS patch antenna resonating at 1.5925 GHz are developed. This work presents the design, modelling, determining passive gain of the GPS patch vs. GNSS antenna with intended targeted applications within the automotive system. Simulation are undertaken to evaluate the performance of the proposed GNSS antenna. Simulation conducted in FEKO software rather than mathematical modelling. The two antennas are also compared from the size standpoint. The goal of this paper is to test, measure and evaluate the performance of GPS against GNSS antennas. Another emphasis of this paper is how to obtain the equivalent amount of total passive gain in a GPS vs. that of GNSS antenna.

KEYWORDS

Differential Global Position System (dGPS), Global Navigation Satellite System (GNSS), Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS), Advanced Driver Assistance Systems (ADAS), Automated Driving (AD), Modelling, comparison, measurements, analysis

1. INTRODUCTION

An antenna could be defined as a wireless communication device or module such as a piece of wire for radiating or receiving electromagnetic wave propagating in a communication channel, such as guided structure transmission line and then getting transmitted into a free space and/or vice versa in the receiving mode. We present the passive gain of the GPS only patch and GNSS (GPS/GLONASS) antenna structure using FEKO electromagnetic simulation software package, in order to support automotive applications. This study describes the modelling, design, simulation and analysis of GPS only (L1) patch and GNSS (GPS/GLONASS) patch antenna. According to Constantine A. Balanis, the antenna is the transitional structure between free-space and a guiding device, for wireless communication systems, the antenna is one of the most critical components. For the past few decade Microstrip Patch Antenna were used heavily in high performance aircraft, spacecraft, satellite and missile where size, weight, cost, performance, ease

of installation, and aerodynamic profile are constraints. Low profile antennas maybe required [1] for packaging and/or aesthetic constraints. The GNSS (GPS/GLONASS) antennas play a significant role in today's modern communications, i.e. they nicely meet automotive specification requirements, most antenna designers and OEMs mainly preferred and select this rectangular/square GNSS (GPS/GLONASS) patch antenna, in order to mount, install, place and position it on their production vehicles.

2. ANTENNA PERFORMANCE CHARACTERISTICS

In the following below are illustration of mathematical model/equations that define the antenna gain, efficiency, directivity and input impedance.

Equation 1 Expresses, the antenna gain as:

$$\text{Gain } G = 4(\pi)(\text{radiation intensity})/\text{total input(accepted) power} = 4(\pi) * (U(\text{Elevation Angle, Azimuth Angle})) / P_{in} \text{ (dimensionless)} \quad [1] \quad (1)$$

Also, Antenna Gain (G) = Antenna Efficiency (ϵ) * Antenna Directivity (D) [2]

Equation 2 Expresses the antenna Radiation efficiency:

$$\epsilon = P_{rad} / P_t \text{ (dimensionless)} \quad (2)$$

Where P_{rad} = Radiated Power, P_t = Transmitter Power [2]

In general, the overall antenna efficiency can be express as below

$$\epsilon_o = \epsilon_r \epsilon_c \epsilon_d$$

Where ϵ_o is the total efficiency (dimensionless)

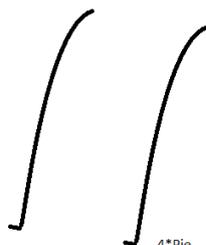
ϵ_r = reflection (mismatch) efficiency = $(1 - |\text{voltage reflection coefficient at the input terminals of the antenna}|^2)$ (dimensionless)

ϵ_c = conduction efficiency (dimensionless)

ϵ_d = dielectric efficiency (dimensionless) [1]

Equation 3 Expresses, the antenna directivity

$$D = 4(\pi) / \Omega \quad (3)$$



Where Ω = Pattern Solid Angle = $\int_{4\pi} F(\text{Elevation Angle, Azimuth Angle}) d\Omega$
 $F(\text{Elevation Angle, Azimuth Angle})$ = Normalized Radiation Intensity [2]

Equation 4 Highlights, the antenna input impedance, defined as:

$$\text{Input Impedance} = Z_A = R_A + jX_A \text{ (ohms)} \quad [1] \quad (4)$$

Where

Z_A = antenna impedance at the input terminals of an antenna when it operates in transmitting mode (ohms)

R_A = antenna resistance at the input terminals of an antenna when it operates in transmitting mode (ohms)

X_A = antenna reactance at the input terminals of an antenna when it operates in transmitting mode (ohms)

In general, the R_A parameter from

Equation 5 is mainly made up of two resistances (R_r and R_L) of the antenna
 Resistive component = $R_A = R_r + R_L$ (ohms) [1] (5)

Where

R_r = Represents the radiation resistance of the antenna (ohms)

R_L = Represents the loss resistance of the antenna (ohms)

If we assume that the antenna is connected/attached to a signal/function generator/source with internal impedance, when the antenna is used in the transmitting mode of operation then internal impedance is defines as listed below:

$$\text{Internal impedance} (Z_g) = R_g + jX_g \text{ (ohms)} \quad [1] \quad (6)$$

Where

R_g = Represents the resistance of signal source/generator impedance (ohms)

X_g = Represents the reactance of signal source/generator impedance (ohms)

Solving these equations at high level will allow to obtain the some of the antenna characteristics. Where on the other hand FEKO simulator/simulation software package is based on the Method of Moments (MoM) integral formulation of James Maxwell's equations, in order to solve for antenna characteristics, such as antenna gain, antenna input impedance, etc.

3. ANTENNA DESIGN STRUCTURE AND ANTENNA EVALUATION

The testing, experimental, comparison and evaluation of the square GPS only patch and GNSS (GPS/GLONASS) patch antenna design and simulation of the proposed two antennas is performed using FEKO which has not been previously investigated and/or studied at the FEKO simulation level. The GPS only patch and GNSS (GPS/GLONASS) patch antenna will be compared and contrasted mostly from the total passive gain in FEKO simulation environment viewpoint. The photo of each of the respective two test and presented antennas under evaluation and assessment are outlined and shown in Figure 1, Figure 2, Figure 3, and Figure 4. Figure 1 depicts the front view of the dual band GNSS (GPS/GLONASS) antenna. From the Figure 2 we

can see the top view of the dual band GNSS (GPS/GLONASS) antenna. Figure 3 depicts the front view of the GPS only L1 square/rectangular patch antenna. From the Figure 4 we can see the top view of the GPS only L1 square/rectangular patch antenna.

3.1. GNSS (GPS/GLONASS) Patch Antenna Front View Photo



Figure 1. Dual Band Constellation GNSS Patch Antenna

3.2. GNSS (GPS/GLONASS) Patch Antenna Top View Photo



Figure 2. Dual Band Constellation GNSS Patch Antenna

3.3. GPS Only Passive Patch Antenna Front View Photo



Figure 3. GPS Only Patch Antenna

3.4. GPS Only Passive Patch Antenna Top View Photo



Figure 4. GPS Only Patch Antenna

3.5. GPS Antenna Samples Consideration for the Evaluation

Table 1. One sample of each antenna used for the model simulation in the FEKO environment

Dual-Band GNSS (GPS/GLONASS) Patch Antenna	1
GPS Only Passive Patch Antenna	1

3.6. Range of Operating and Simulation Frequency Requirements

The GNSS (GPS/GLONASS), GPS Only L1 frequencies were used for the purpose of this paper test and simulation activities:

- GPS Only L1 Frequency, GPS (L1): 1.57542 GHz
- GNSS (GPS/GLONASS): 1575 to 1610 MHz, Center Frequency (f_c)= 1.5925 GHz)

The proposed antennas [GPS Passive Patch and GNSS (GPS/GLONASS)] characteristic was simulated by using FEKO simulation software package. An analysis was conducted next and finally total gain for each of the sample antenna were observed from the POSTFEKO environment.

4. FEKO DESIGN PARAMETERS AND SIMULATION RESULTS

4.1. Antenna Samples for FEKO Simulation

Table 2. Antenna substrate and radiating element dimensions

Patch Size & Application	Reference & Device Under Test (DUT) Antenna
GPS Only Single Passive Patch Antenna (L1-, 1.57542 GHz) <ul style="list-style-type: none"> • Substrate Size: 24.9 x 24.8 x 4.5 mm • Radiating Element Size: 12.25 x 12.25 mm 	Reference
Dual Band GPS/GLONASS Antenna (GNSS-1.5925 GHz) <ul style="list-style-type: none"> • Substrate Size: 24.7 x 24.7 x 4.5 mm • Radiating Element Size: 12.25 x 12.25 mm 	DUT

4.2. Design Parameters within FEKO Simulation Environment

Table 3. FEKO Mesh and Loss Tangent Parameters for GPS only Patch and GNSS Patch Antenna

GPS and GNSS Antenna Component	Parameter	Value
GPS only Patch Antenna Operating at , 1.57542 GHz	Mesh-Wire Segment Radius	1.587e-3 mm
	Dielectric Loss Tangent for Porcelain Material	2.1e-14 mm
GNSS (GPS/GLONASS) Patch Antenna Operating at 1.5925 GHz	Mesh-Wire Segment Radius	1.569e-3 mm
	Dielectric Loss Tangent for Porcelain Material	2.0e-14 mm

4.3. Reference GPS only L1 Patch Antenna

Modelling, design, and simulation based on the following design parameters listed in Table 4 below.

Table 4. Design Parameters of Reference GPS only Structure Antenna

Parameter	Value
Feed Length	0.5 mm
Operating Frequency	1.57542 GHz
Ground Plane Length	95 mm
Ground Plane Width	95 mm
Radiating Element Length	12.25 mm
Radiating Element Width	12.25 mm
Substrate Length	24.8 mm
Substrate Width	24.9 mm
Substrate Thickness	4.5 mm
Substrate Dielectric Constant (Relative Permittivity) for Ceramic/Porcelain Material	5.5 mm

Figure 5 plot below shows graphic representation of a pin fed voltage source of excitation. The Top View of the GPS only (L1 frequency, 1.57542 GHz) passive patch antenna on a finite

square/rectangular orange color ground plane, square purple color substrate with dielectric constant value of 5.5 mm and a square dark blue color radiating element. Figure 6 depict the Side View of the GPS only patch antenna.

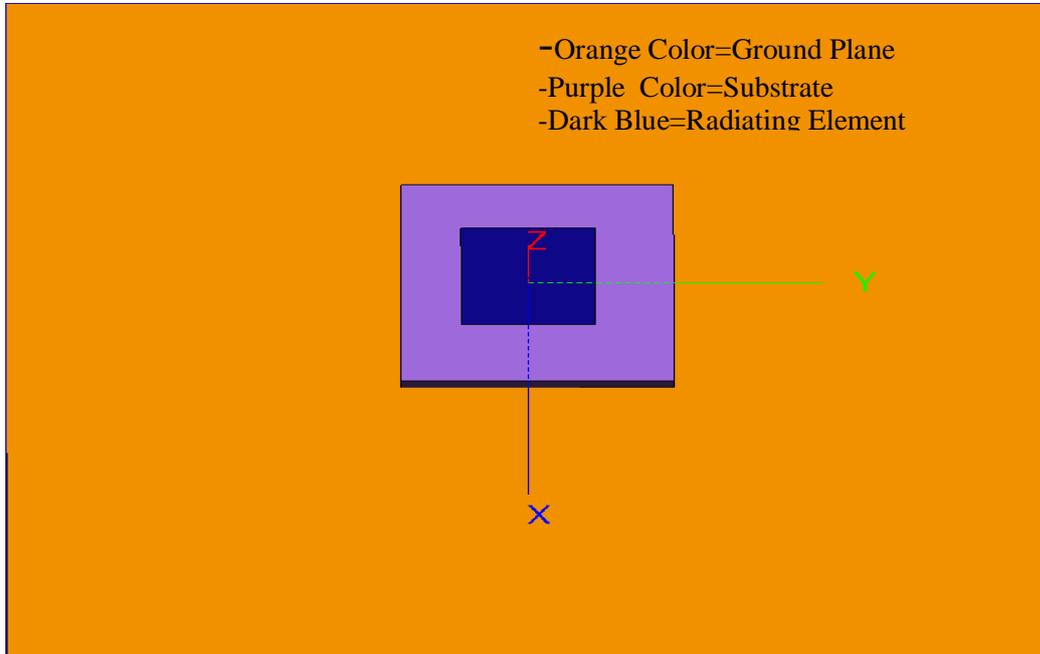


Figure 5. GPS Patch antenna operating at 1.57542 GHz (Top View/Cross Section Image)

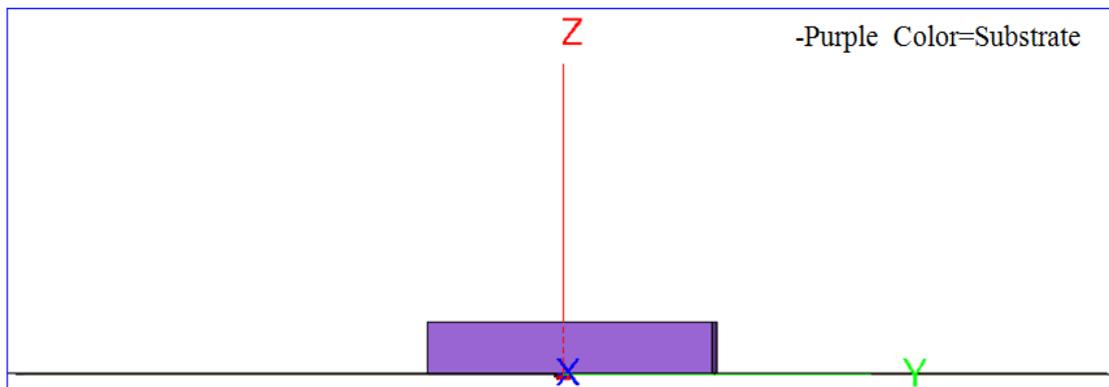


Figure 6. GPS Patch antenna operating at 1.57542 GHz (Side View)

4.4. Simulated Far Field of Reference GPS only Structure Patch Antenna

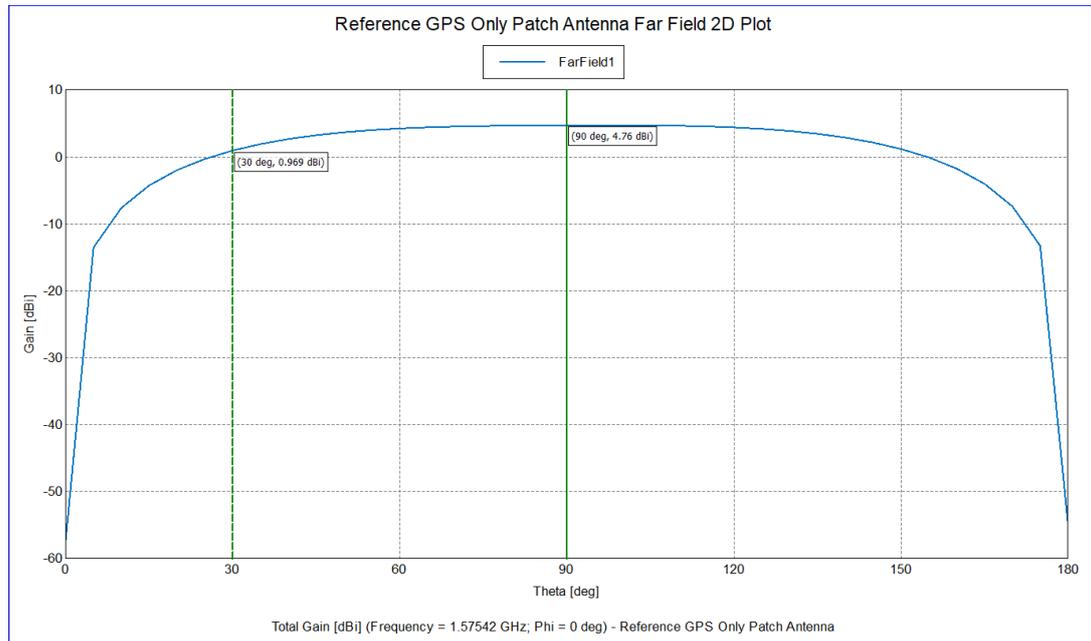


Figure 7. Patch antenna operating at 1.57542 GHz

Using Figure 7, the passive gain is approximately 3.791 dBi of the presented antenna and it can be determined by taking the difference in gain angle/delta between 30 and 90 degree angles in the 2D plot graphic.

Figure 7 shows the simulated passive gain of the proposed antenna and I selected substrate material to be Ceramic/Porcelain with dielectric constant/relative permittivity=5.5 mm to model, design, and simulate the presented GPS (L1) only patch antenna.

4.5. Device Under Test (DUT) GNSS Patch Antenna

Modelling, design, and simulation based on the following design parameters listed in table 5 below.

Table 5. Design Parameters of Device Under Test (DUT) GNSS (GPS/GLONASS) Structure Patch Antenna

Parameter	Value
Feed Length	0.5 mm
Operating Frequency	1.5925 GHz
Ground Plane Length	95 mm
Ground Plane Width	95 mm
Radiating Element Length	12.25 mm
Radiating Element Width	12.25 mm
Substrate Length	24.7 mm
Substrate Width	24.7 mm
Substrate Thickness	4.5 mm
Substrate Dielectric Constant (Relative Permittivity) for Ceramic/Porcelain Material	5.5 mm

Figure 8 plot shows the Top View of the GNSS (GPS/GLONASS) patch antenna with a square/rectangular orange color ground plane, square purple color substrate with dielectric constant value of 5.5 mm and a square dark blue color radiating element. Fig 9 depict the Side View of the GNSS (GPS/GLONASS) patch antenna.

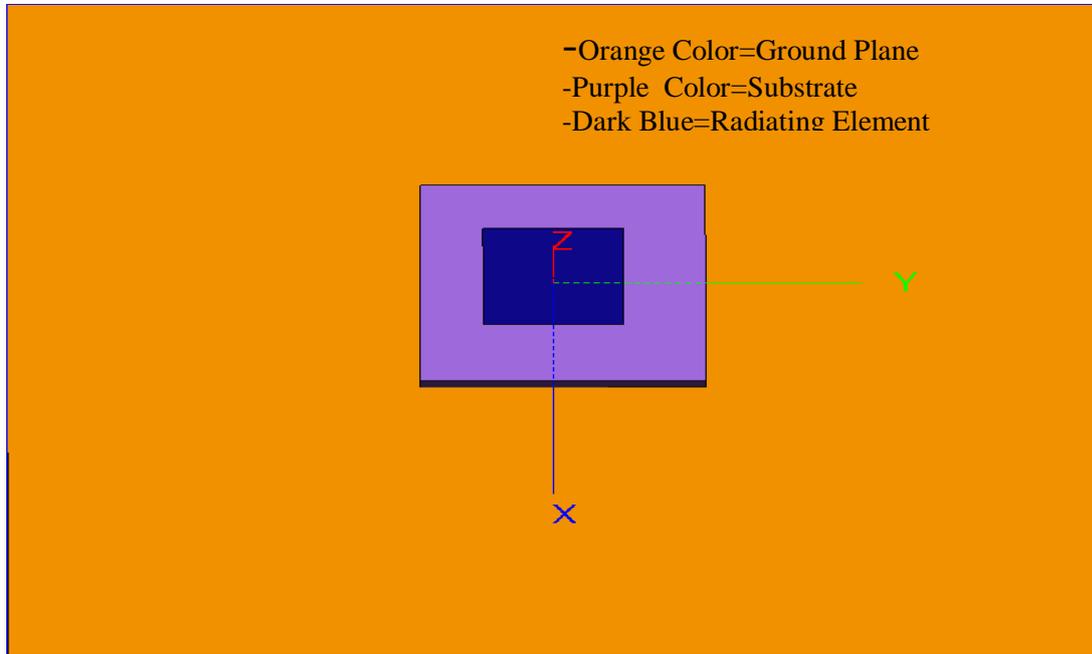


Figure 8. DUT GNSS Patch antenna operating at 1.5925 GHz (Top View/Cross Section Image)

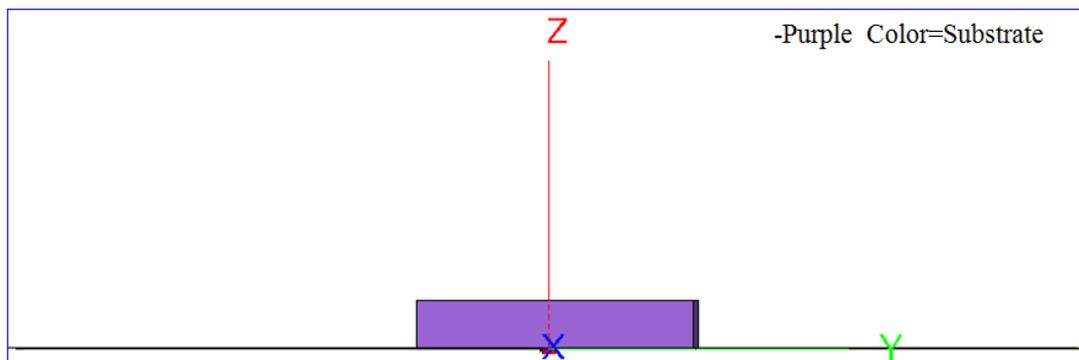


Figure 9. DUT GNSS Patch antenna operating at 1.5925 GHz (Side View)

4.6. Simulated Far Field of DUT GNSS Structure Patch Antenna

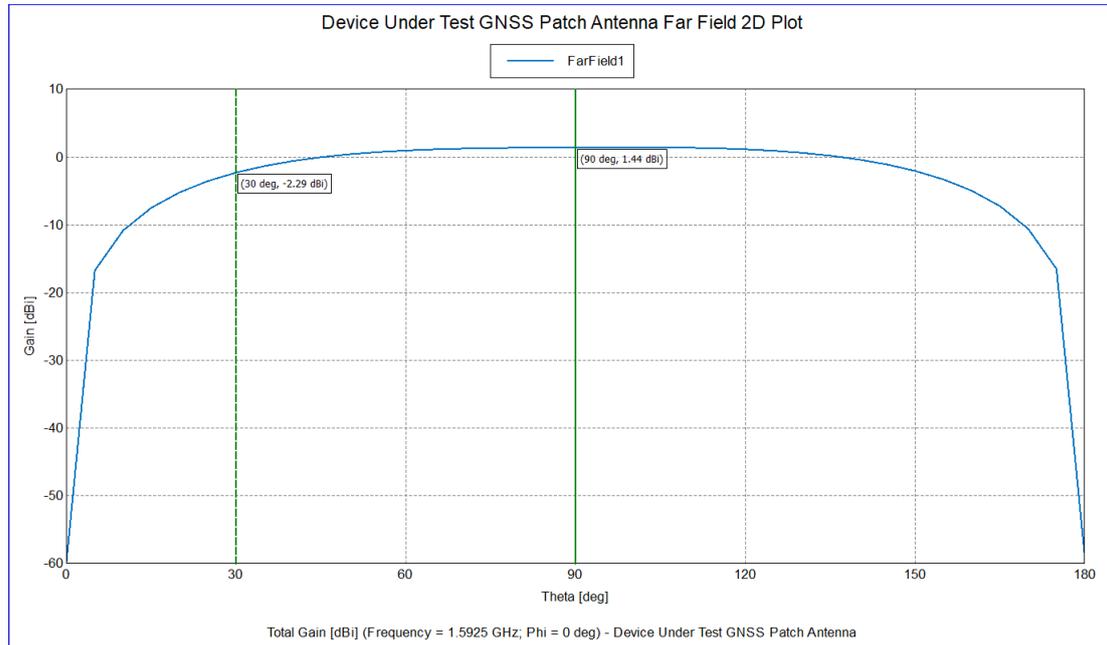


Figure 10. Patch antenna passive gain at 1.5925 GHz

In the 2D model plot from figure 10, we can see about 0.85 dBi passive gain, by taking the difference in gain between 30 and 90 degrees.

Figure 10 simulated the passive gain of the proposed antenna and we selected substrate material to be Ceramic/Porcelain with dielectric constant/relative permittivity=5.5 mm to model, design, and simulate the presented GNSS (GPS/GLONASS) patch antenna.

5. CONCLUSIONS

This paper describes the GPS (L1 1.57542 GHz) frequency) patch antenna performance and compares it to that of GNSS patch antenna with ceramic/porcelain substrate material. These two antennas can be used in modern automotive applications. The models for each antenna were developed and then simulated on FEKO. The performance characteristic, such as passive gain in dBi were found, 3.791 dBi for GPS and 0.85 dBi for GNSS delta between 30 and 90 degrees. The simulated results show an improved passive gain for the antenna. Thus, the proposed GNSS will meet the needs of future automotive applications in robust way. Furthermore, other characteristics such as wide band width and efficiency are examined.

6. FUTURE WORK SUGGESTIONS

In future work this presented antenna can be modified, further studied and simulated in each of the following manners:

- The substrate material type can be changed from ceramic/porcelain to non-ceramic version/variant.

- The proposed antenna performance can be further improved by selecting a thick substrate whose relative permittivity is in the lower/smaller value than the presented dielectric constant of 5.5 mm.
- The mechanical dimensions of each antenna can be altered and simulated enhance antenna major performance parameters, such as Directivity, Impedance, Current and Polarization.
- The verification, validation and testing of the proposed GPS only patch and GNSS patch antenna component can also be conducted at the system vehicle level, where each of the presented antenna can be installed and mounted on an optimal vehicle roof location area prior to the start of the testing and antenna performance parameters experimental measurement can be ascertained.
- The testing, assessment and evaluation of the presented GPS only (L1 1575.42 MHz frequency) and GNSS (GPS/GLONASS) patch antenna can be carried out in anechoic chamber and/or indoor antenna range, in order to measure the basic antenna performance parameters and/or characteristics, such as radiation pattern, radiation efficiency, directivity, impedance, polarization and current draw.

REFERENCES

- [1] Constantine A. Balanis, Antenna Theory Analysis and Design, 2nd ed., New York: Wiley, 1997, p. 1-3, 58-61, 73-77, 722-750, 839-871
- [2] Fawwaz T. Ulaby, Fundamentals of Applied Electromagnetics, 2001 Media Edition, Prentice Hall, p.347-354

AUTHORS

Gholam Aghashirin graduated from Ryerson University, Toronto, Ontario Canada with a B.Eng. in Electrical, Electronics and Communication Systems, earned his M.Sc. in Electrical and Computer Engineering from Oakland University, Rochester, Michigan, USA and he is currently a Ph.D. candidate in Electrical and Computer Engineering at Oakland University, Rochester, Michigan, USA. He has worked as an Engineer in advanced engineering projects, assignments in the automotive industries at various level of complexity and leadership roles in the field and space of Global Telematics, Automotive Radio Head Units, Navigation Systems, Instrument Clusters, Voice Recognition, Dialog, Hands-Free Systems, Electrical and Electronics ADAS and Automated Driving Systems. His research interests include Electromagnetics, location technologies, antenna design, modelling, simulations at the component, vehicle level, and antenna experimental measurements.



Hoda S. Abdel-Aty-Zohdy received the B.A.Sc. degree (with First Class Honors) in Electrical and Communications Engineering from Cairo University, the M.A.Sc and Ph.D. degrees in Electrical Engineering from the University of Waterloo, ON, Canada. Dr. AbdelAty-Zohdy is a Professor of Electrical and Computer Engineering, The John F. Dodge Chair Professor of Engineering, 2012-2014; Director of the Microelectronics & Bio-Inspired Systems Design Lab at Oakland University, Rochester, MI, USA. Her research and teaching focus on Circuits, Devices, VLSIC, H/W deep-learning, Electronic-Nose, and Bio-Inspired IC chips for high fidelity classifications. She organized, chaired, served on several conferences and committees for the IEEE/CASS and as Distinguished Lecturer 2004-2006.



Adam Timmons received the Ph.D. degree in Materials Science from Dalhousie University, Halifax, Nova Scotia, Canada. Dr. Adam is an Adjunct Professor within the Department of Mechanical Engineering at McMaster University, Hamilton, ON, Canada. He has many professional and academic appointments and holds a large number of patents.



Mohamed A. Zohdy received the B.A.Sc degree in Electrical Engineering from University of Cairo, the M.A.Sc and Ph.D. (Medal) from the University of Waterloo, ON, Canada. Dr. Mohamed is a Professor of Electrical and Computer Engineering at Oakland University, Rochester, MI, USA. Professor Mohamed research focus is in the area of Advanced control and estimation, intelligent pattern information processing, neural, fuzzy, evolutionary systems, chaos control, smart simulation, hybrid systems.



© 2020 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

Classification of Fatigue in Consumer-grade EEG Using Entropies as Features

Muhammad Azam, Derek Jacoby, and Yvonne Coady

Department of Computer Science, University of Victoria,
Victoria, Canada

Abstract. Electroencephalogram (EEG) records electrical activity at different locations in the brain. It is used to identify abnormalities and support the diagnoses of different disease conditions. The accessibility of low-cost EEG devices has seen the analysis of this data become more common in other research domains. In this work, we assess the performance of using Approximate entropy, Sample entropy, and Reyni entropy as features in the classification of fatigue from EEG data captured by a MUSE 2 headset. We test 5 classifiers: Naive Bayes, Radial Basis Function Network, Support Vector Machine, K-Nearest Neighbor, and Best First Decision Tree. We achieved the highest accuracy of 77.5% using the Support Vector Machine classifier and present possible enhancements to improve this.

Keywords: EEG, Electroencephalogram, Approximate entropy, Sample entropy, Reyni entropy, Fatigue detection, Automatic classification, MUSE 2

1 Introduction

Fatigue has become a prominent issue in our society and more people are starting to suffer from ‘burnout’. It is defined as a physiological state of reduced mental or physical capability [1]. When a person experiences a fatigue condition and is operating at a reduced capability, this can lead to accidents and mistakes which can have very real consequences. This is apparent in fatigue-related traffic accidents, where research has shown that 24 hours of prolonged wakefulness has similar adverse effects on cognitive performance to a blood alcohol level of 100 mg per 100 ml of blood [2]. This exceeds the legal limit in British Columbia, Canada by 25%.

Hospitals are known to be stressful and demanding work environments. Staff often work long shifts, which can sometimes be back-to-back. The effects of reduced cognitive and physical performance in such a workplace, can have serious consequences for patient health, especially as workers experiencing constant acute fatigue will likely exhibit more residual fatigue over time.

Yet with all of the adverse effects, fatigue is not monitored, diagnosed, or treated in the same way as other health risks such as obesity or diabetes. Part of the reason for this could be determining and monitoring it is not straightforward. For diabetics, test strips provide a fast and convenient way to measure blood sugar

levels, allowing for effective management. For patients experiencing obesity, there are numerous ways of measurement, and well researched practices for weight loss.

For fatigue, there is no simple device that one can go out and buy, and no straightforward method that can be used at home to help manage it. In order for such a device to be effective, it would be critical that it not be invasive, be able to accurately detect fatigue levels, track trends, and perhaps be adaptable for a treatment plan.

This work looks at addressing the detection component. The first step is to create a computerized and automated process that can consume EEG data and provide a binary classification of whether the condition is fatigued or normal. If successful, this methodology may be adaptable to determine not only the current condition, but also the level of fatigue accumulation. A reduction in cognitive performance would likely alter the electrical activity in the brain, making EEG signals ideal for this analysis. In addition, non-clinical devices such as the MUSE 2 headset used to collect the data used in this work, are becoming more common and cost-effective to acquire.

The approach taken adapts a method used in [3] to determine focal and non focal EEG signals for identification of epileptic disease conditions using entropies.

This paper is organized as follows: Section 2 provides references to related work that were consulted, Section 3 provides a detailed description of the data used, a brief overview of the entropies used as features, as well as some information about the classifiers used in the training process. Section 4 presents the results that were obtained through this process. Section 5 provides a discussion on the performance, as well as possibilities for improving the classification process. Finally, the conclusion is given in Section 6.

2 Related Work

To determine an effective approach in detection of fatigue in EEG signals, a number of works aimed at producing features from EEG signals and classifying them were consulted. In [3], three entropy features, Approximate entropy, Sample entropy, and Reyni entropy, were extracted from the Bern Barcelona EEG dataset, and 5 classifiers were trained using a subset of the input data to detect focal and non-focal EEG nodes to detect underlying epileptic conditions. The performance of the classifiers was then tested with novel data and determined to be over 90%, with the best performance of 98% being for a Non-Nested Generalized Exemplars (NNGE) classifier. The dataset used in [3] is described in detail in [4]. Srinivasan et al. have also proposed using Approximate entropy, but in conjunction with artificial neural networks to classify epileptic conditions in EEG data [5]. Other methods using neural networks [6] as well as signal decomposition techniques [7] have also been used in epilepsy detection.

Further examples of detecting epileptic conditions in EEG are also available. In [8], Acharya et al. perform analysis on EEG data using Approximate entropy, Sample entropy, and Phase entropies (S1 and S2) as features. Additional classifiers are used in this work, with the highest performance in accuracy of 98.1% being attributed to the Fuzzy Sugeno Classifier. Research has also been conducted on quantifying depth of anesthesia [9] as well as coma [10] and unresponsive [11] states, using a combination of EEG data and extracted entropy features.

In the domain of sleep stage classification, Rodríguez-Sotelo et al. [12] show that unsupervised learning techniques to automatically classify sleep stages using Shannon entropy, Approximate entropy, Sample entropy, and Multiscale entropy (in addition to a number of features that draw from the complexity of a signal) work well. A comparison of classification accuracy on the features showed that Approximate and Sample entropy performed at 74% and 73% respectively.

Driver fatigue detection and correction is also a common application of fatigue detection in EEG data. Huang et al. [13] go a step beyond the detection of fatigue in simulated driving conditions, and at the onset of detected fatigue, provide a warning signal to the driver. This has the effect of improved behavioural performance following the signal, but a reduction in efficacy is also shown with repeated signals. To detect the onset of fatigue, the authors analyze the EEG power (alpha band) and a high value exceeding a pre-determined threshold signifies that the driver is approaching a fatigue state. The threshold was determined experimentally through baseline measurements for each participant.

Application of wireless headsets for detecting and providing real-time feedback on high-speed train driver fatigue are shown in [14]. The method presented uses a Fast Fourier Transform (FFT) feature extraction combined with a Support Vector Machine (SVM) classifier to feed the detection model. Alerts are then presented if the onset of fatigue is detected. Classification accuracy is shown to achieve between 77% and 96% accuracy. FFT features are widely used in the literature to extract features from signals for classification models.

Much of the literature showcases examples of classification using different techniques/features but with the use of medical grade EEG devices. In this work, we evaluate the use of entropy as features in automatically detecting a fatigue condition using a device that has limited nodes and configuration options, but is more suitable for general at-home use. Our goal is to evaluate whether reliable enough performance can be achieved, that make the MUSE 2 headset a viable fatigue tracking tool.

3 Proposed Approach

This section describes the dataset used for this work, the features extracted, as well as a brief overview of the classifiers used.

3.1 Data

The data used for this work was obtained from Krigolson Labs at the University of Victoria [15]. The data was captured using a MUSE 2 headset from participants before and after working a 12 hour shift in a hospital environment. The data collection was performed in 2018 for 15 participants, and in 2019 for 33 participants (2 observations are missing in the dataset for the fatigue condition for this year). EEG recordings taken with the headset before the shift are ground truth labeled as ‘normal’ condition, and recordings after the shift was complete were classified as ‘fatigue’ condition. The recordings all vary in length from 41,988 - 129,132 samples. At the sampling frequency of 256 Hz, these recordings fall between approximately 2.7 - 8.4 minutes of signal data. The MUSE 2 headset has 4 electrodes and, since these are fixed in the device with no options for different device sizes, the locations of readings are likely non-standard [16].

The raw data was contained in .csv format and an example signal for normal condition is shown in Figure 1. Preprocessing was required before data was appropriately usable, including header corrections, fixing junk values, format transformations, and mean removal. No noise removal or filtering techniques were applied to the data.

An additional step of data preparation was necessary before the feature extraction process. This involved converting each 4 channel signal into 4 individual signals. In related literature, readings from the electrode of an EEG device are often taken as individual signals for classification purposes. An example of this is shown in [3]. Segmentation was also applied to the data to transform the signals into a fixed length of 10,240 samples, or 40 seconds at the sampling frequency. Each signal that was divided into the subsets also had the same ground truth label applied. The effect of this preparation step greatly increased the dataset size to 1,244 signals of fatigue condition and 1,360 signals of normal condition.

3.2 Feature Extraction

In machine learning, pattern recognition, and image processing tasks, feature extraction is a key component to training a system well so that it is able to recognize and classify novel data accurately. Features are mathematical descriptors of the data that lend well to the application of classifier networks that use statistical techniques, or in the more complex case, deep learning.

Some simpler features that can be extracted from signals include mean, standard deviation, minimums, and maximums. The frequency information of a signal can also be used to extract quality features. It is often the case that raw data values are simply too large to process. For example, in the case of each signal used here there would be 10,240 features, so through careful feature extraction and selection the number of features describing our data is reduced down to just three. In this work,

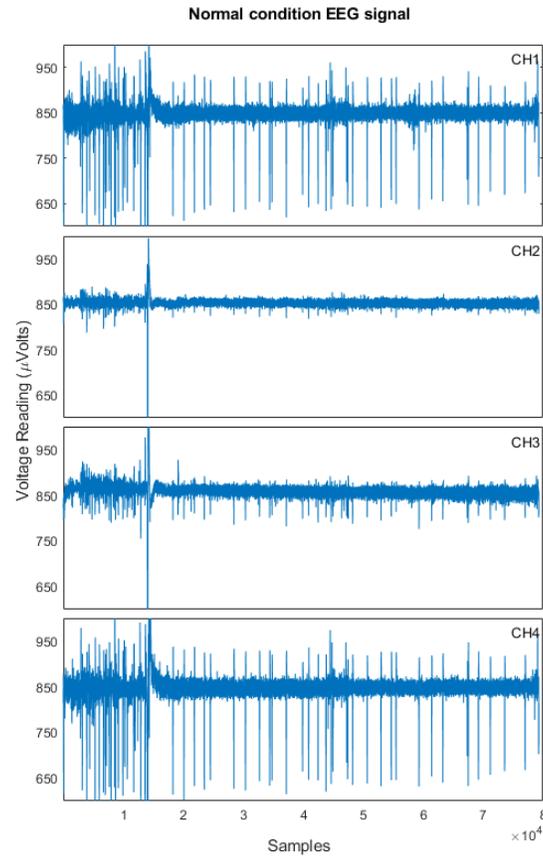


Fig. 1. Normal condition raw signal data

we will use three entropies as features of the data: Approximate entropy, Sample entropy, and Reyni entropy. Entropy have been used extensively in the literature to perform classification of signals in various domains.

3.3 Approximate entropy

Approximate entropy is a statistical technique that is used to measure the regularity of a signal. Low Approximate entropy is a reflection of the persistent, repetitive, and predictive nature of a series with apparent patterns that repeat themselves [17]. The value of using this entropy is that it can distinguish between series that have the same moment statistics (mean, standard deviation, etc.) but may be more different in random probability. A completely random and unpredictable signal will produce a high value of Approximate entropy, i.e. it is chaotic. A more regular

and predictable series will yield a low value. To calculate this value, an integer m defines the length of the window size and r specifies the comparison filtering level. The algorithm to determine the approximate entropy for a given time series is as follows:

Step 1: Form a time series of data with length N :

$$u(t) = u(1), u(2), u(3), \dots, u(N) \quad (1)$$

Step 2: From the time series, form a sequence of vectors each of length m :

$$x(\vec{1}), x(\vec{2}), x(\vec{3}), \dots, x(N - \vec{m} + 1)$$

where

$$x(\vec{1}) = \begin{bmatrix} u(1) \\ u(2) \\ \dots \\ u(m) \end{bmatrix}, x(\vec{2}) = \begin{bmatrix} u(2) \\ u(3) \\ \dots \\ u(m+1) \end{bmatrix}, x(\vec{3}) = \begin{bmatrix} u(3) \\ u(4) \\ \dots \\ u(m+2) \end{bmatrix} \dots$$

and generally

$$x(\vec{i}) = \begin{bmatrix} u(i) \\ u(i+1) \\ \dots \\ u(i+m-1) \end{bmatrix} \quad (2)$$

Step 3: Calculate the maximum distance between a vector $x(\vec{i})$ and all other $x(\vec{j})$ in the set to create a distance value vector:

$$d(\vec{i}) = \begin{bmatrix} \max(\text{dist}|x(\vec{i}), x(\vec{1})|) \\ \max(\text{dist}|x(\vec{i}), x(\vec{2})|) \\ \dots \\ \max(\text{dist}|x(\vec{i}), (x(N - \vec{m} + 1))|) \end{bmatrix} \quad (3)$$

Step 4: Calculate the number of values in each distance vector that are within the filtering criterion m to get a ratio C for each:

$$C_i^m(r) = \frac{\sum (d(\vec{i}) \leq m)}{N - m + 1} \quad (4)$$

Step 5: Define ϕ as the negative logarithmic sum of ratio values:

$$\phi^m(r) = \frac{\sum_{i=1}^{N-m+1} \log C_i^m(r)}{N - m + 1} \quad (5)$$

Step 6: Increase m by 1 and repeat the process to get:

$$\phi^{m+1}(r)$$

Step 7: Approximate entropy (AE) is defined as the difference of these two:

$$AE(m, r) = \phi^m(r) - \phi^{m+1}(r) \quad (6)$$

Typically, the initial value of m is taken as 2 or 3, and the value of r greatly depends on the data and application domain. In this work, we set m equal to 2, and the filtering parameter r equal to 3. A full detailed description and analysis of Approximate entropy is provided in [17].

There are two important implications to using Approximate entropy: the first is that relative consistency is not guaranteed as a different filtering parameter of r may yield entirely different results. The second is that this entropy calculation depends heavily on the length of the series. Approximate entropy is also biased towards suggesting more regularity than there actually is in the signal.

3.4 Sample entropy

Sample entropy is a modification of Approximate entropy that addresses the issues mentioned above with Approximate entropy. This statistical technique excludes self-counting, maintains relative consistency and is mostly independent of the length of the series [17].

The Sample entropy algorithm requires as input a window size m and filtering parameter r . It is calculated through the following steps:

Step 1: Form a time series of data with length \mathbf{N} :

$$u(t) = u(1), u(2), u(3), \dots, u(N) \quad (7)$$

Step 2: Form two sets of vectors as follows:

$$x_m^{\vec{i}} = \begin{bmatrix} u(1) & \dots & u(N-m) \\ \dots & \dots & \dots \\ u(m) & \dots & u(N-m+1) \end{bmatrix}$$

$$x_m^{\vec{j}} = \begin{bmatrix} u(1) & \dots & u(N-m+1) \\ \dots & \dots & \dots \\ u(m) & \dots & u(N) \end{bmatrix}$$

Step 3: Calculate the number of vector pairs at a distance less than r as:

$$B = \left[\sum_1^{N-m} \text{count}(\text{dist}|x_m^{\vec{i}}, x_m^{\vec{j}}|) < r \right] - 1 \quad (8)$$

Step 4: Increase m by 1, and repeat the process to get:

$$A = \left[\sum_1^{N-m+1} \text{count}(\text{dist}|x_{m+1}^{\vec{}}(i), x_{m+1}^{\vec{}}(j)|) < r \right] - 1 \quad (9)$$

Step 5: The sample entropy (SE) is then defined as:

$$SE(m, r) = -\log \frac{A}{B} \quad (10)$$

Both A and B represent probabilities that two sequences are similar. Detailed analysis and comparison of Approximate and Sample entropy is provided in [17].

3.5 Reyni entropy

Reyni entropy (RE) is a generalization on a number of other entropies, specifically the Hartley entropy, Shannon entropy, the collision entropy, and the min-entropy. It is given by:

$$RE(\alpha) = -\frac{\alpha}{1-\alpha} \sum \log p_k^\alpha \quad (11)$$

where

$$\alpha \geq 0, \alpha \neq 0$$

With specific values of α , Reyni entropy simplifies to one of the above mentioned types.

3.6 Classification

Once features have been extracted from the data, we use a subset to train 5 different classifiers. This training process yields a parameter-based model for each that can be used to predict values on novel data. The remaining unused data is then fed into each of the classifiers, and the predicted output labels are matched to ground truth labels to determine performance. A brief overview of the classifiers used is provided in this section.

3.7 Naive Bayes (NB)

The Naive Bayes family of classifiers works by assuming that each of the input features are independent variables: that is, that each input feature is not dependent on any other feature. While this is generally a poor assumption to make since features often have some relationship or dependence describing the problem, these classifiers work surprisingly well in practice. The Bayes model works on conditional probability, determining an outcome class based on some number of independent variables (the input features) [18].

3.8 Radial Basis Function Network (RBF)

A Radial Basis Function is a function whose value depends only on the distance between the input and some fixed point. The RBF network is a simple single-layer artificial neural network where the RBF is used as the activation function. As RBF's operate on distance approximation, the data was modified to represent the fatigue and normal states as absolute distance values. Since the output of this network produces a real-valued distance approximation, results of the classified test data had to be converted back to labels.

3.9 K-Nearest Neighbour (KNN)

This classifier works on grouping incoming data points into the member group of the nearest neighbour. The computation uses distance values to determine which nearest neighbour to group to. Different types of distances can be used, but experimentally we determined Euclidean distance to perform optimally.

3.10 Support Vector Machines (SVM)

Support Vector Machine classifiers work by attempting to draw an optimal hyper-plane that separates clusters of data points belonging to the same member groups (classification types). Support vectors are then drawn around the plane. The performance of this classifier is highly dependent on if the drawn hyper-plane can effectively separate points representing different classifications, and by the width of the support vectors around this plane. A larger width means that the data is effectively separable.

3.11 Best First Decision Tree (BFDT)

A decision tree is a set of conditional control statements where each node represents a "test" on an attribute. The branch from each node represents the outcome of the node. The paths from root to leaf (end node) represent the classification rules used in this classifier [19].

3.12 Performance Measures

Performance on novel test data for each of the classifiers is determined by 6 different measures. These measures are calculated from Type I and II errors (false positive (FP) and negative(FN)) as well as true positive (TP) and negative (TN) predicted outcomes. Here, positive refers to the fatigue condition and negative refers to the normal condition. The measures are:

Accuracy (ACC):

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

Sensitivity (SEN):

$$\frac{TP}{TP + FN} \quad (13)$$

Specificity (SPF):

$$\frac{TN}{TN + FP} \quad (14)$$

Positive Predictive Value (PPV):

$$\frac{TP}{TP + FP} \quad (15)$$

Negative Predictive Value (NPV):

$$\frac{TN}{TN + FN} \quad (16)$$

Matthews Correlation Coefficient (MCC):

$$\frac{(TP * TN) - (FN * FP)}{\sqrt{(TP + FN)(TP + FP)}\sqrt{(TN + FN)(TN + FP)}} \quad (17)$$

4 Experimental Results

Calculations of Approximate, Sample, and Reyni entropy values were obtained directly from the prepared 2,604 signals each of length 10,240 samples. Each of resulting 3 x 1 feature vectors per signal had a corresponding ground truth label of 'Normal' or 'Fatigue' condition attached. This was the final state of the data before the training process. To train the classifiers, the input data rows were randomized, and separated into a training set comprising 80% of the data, and a test set which had the remaining 20%.

Each of the classifiers was given the same set of randomly prepared training data, and hyper-parameter optimization was used to determine the best performing parameter settings. This process yielded a trained model for each classifier. Finally, each of the classifiers was given the 3 x 1 feature vectors from the earlier prepared test data, yielding predicted labels for each. These predicated labels were compared to the ground truth values for that data to determine the level of performance. Resulting performance of each classifier is shown in Table 1.

Table 1
Results of classifiers in detecting 'Fatigue' condition

Classifier	ACC	SEN	SPF	PPV	NPV	MCC
Naive Bayes	72.6	46	99.6	99.1	64.4	0.54
Radial Basis Network	70.1	76.7	64.4	64.8	76.4	0.41
Support Vector Machine	77.4	47.5	99.7	99.1	71.7	0.58
K-Nearest Neighbor	77.5	63	90.1	86.3	72.9	0.56
Best First Decision Tree	68.9	53.9	82.5	73.5	66.5	0.38

The results show that the classification accuracy is highest for the SVM and KNN classifiers at about 77%. Values for the other measures are mixed, and show that the classifiers are achieving lower than expected performance.

5 Discussion

In [3], the authors show that using these three entropies as features yields results of over 90% for all of these classifiers in detecting focal and non-focal EEG. That is, EEG channel readings where an underlying epileptic condition exists (focal) and where it does not (non-focal). In theory, this method should apply well to the analysis done here. There are likely a few issues preventing the classifiers from performing better.

The data used for this work was of raw EEG data containing 4 channels. Without additional preprocessing, especially noise removal, it is likely that the entropy algorithms were negatively affected. As shown in Figure 1, significant noise and artefacts appear in the signal which may be removed or suppressed through the application of signal processing techniques. The dataset used in [8] was obtained from the Department of Epileptology at the University of Bonn and artefact free, further providing evidence that EEG data cannot be used in its raw recorded format effectively for feature extraction.

Table 2
Mean and variance of entropy features

Feature	Fatigue	Normal
Approximate Entropy	1.223±0.4823	1.079±0.5581
Sample Entropy	1.148±0.5184	0.951±0.5680
Reyni Entropy	10.85±6.477	13.68±5.669

A statistical analysis of mean and variance of the calculated features is shown in Table 2. As is evident from this, while the mean value is different for each of the conditions in each of the entropy values, the standard deviation is too large

for an effective linearly separable solution. This is likely a major contributor to the performance outcomes seen. When calculating the features, the parameters used can also have significant impact on the outcome value. In the case of Approximate and Sample entropy, while the dimensional parameter is usually kept the same, the filtering parameters depend heavily on the application and signal type. It is possible that the values used during calculations were not appropriate and may require further research to determine.

Data quality issues are likely also present. The MUSE headset operates differently than clinical EEG devices. The headset has fixed contact locations, which may not be ideal for getting optimal readings. The manner in which the headset was worn by the participant could also have affected the recordings. In [16], the authors show some of the challenges involved with data collection using a MUSE headset which can extend to even the amount of hair a participant has affecting readings. Participants themselves could have started the shift already fatigued at the time of the normal condition recording. Different participants may have had varying levels of residual fatigue affecting the observation. Research-grade EEG devices often have many more electrodes available and can be manually adjusted to get optimal readings. This is not the case with the MUSE headset. The utilization of the 4 EEG channels differently could also lead to better results as the process of converting 4 channels to 4 signals required a significant assumption to be made: that electrical activity in the brain should change everywhere under fatigue conditions. This could be a false assumption, and it is possible that certain locations in the brain may be better markers of fatigue than others.

Finally, it is possible that these entropy features are not well suited to detecting fatigue condition from EEG signals. While entropies have been used in the literature to detect various brain conditions such as epilepsy, adaptations may be required for the problem of fatigue detection. Further research using other features such as instantaneous frequency, spectral entropy, and wavelet transformations may yield better results. In [12], power and complexity analysis techniques such as Fractal Dimension actually show a better performance in unsupervised classification tasks. These may also improve the performance of fatigue detection. With the remarkable performance of deep learning networks, using a 2-dimensional spectrogram of the signal in conjunction with a convolutional neural network may also yield better results. Techniques such as this have been performed on audio signals for applications such as music classification [20].

6 Conclusion

Fatigue is becoming more common in society, and as we become more aware in the impacts it has on our daily lives, we are also looking for better ways to manage or reduce it. For non-invasive and easy to use devices that can be used at home, effective detection and measurement of fatigue is critical. In this work, we used

entropy features to train classifiers for detecting fatigue conditions in EEG signals automatically. We achieved the highest performance of 77.5% accuracy using the Support Vector Machine classifier. This performance can likely be improved through better signal processing on the raw data to remove noise and artefacts, as well as by tuning the entropy feature calculations using domain-specific parameters.

Further research on the work presented here may look at binary classification of fatigue using different devices or by comparing the effectiveness of different features. Deep learning techniques may also perform well. Following the development of a binary classification model, the method may be adapted to determine levels of fatigue, given individual baselines. More extensive data collection and individual level baselines may also aid in determining the effect of fatigue on EEG data. Greater accuracy can then be combined with inexpensive devices and connected apps that help individuals monitor and understand their fatigue levels. This can also have significant impacts on stressful work environments where long hours can impact performance.

7 Acknowledgements

The authors would like to thank Krigolson Labs at the University of Victoria for providing the dataset used in this work.

References

1. Lock, A. M., D. L. Bonetti, and A. D. K. Campbell. "The psychological and physiological health effects of fatigue." *Occupational medicine* 68.8 (2018): 502-511.
2. Dawson, Drew, and Kathryn Reid. "Fatigue, alcohol and performance impairment." *Nature* 388.6639 (1997): 235-235.
3. Arunkumar, N., et al. "Classification of focal and non focal EEG using entropies." *Pattern Recognition Letters* 94 (2017): 112-117.
4. Andrzejak, Ralph G., Kaspar Schindler, and Christian Rummel. "Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients." *Physical Review E* 86.4 (2012): 046206.
5. Srinivasan, Vairavan, Chikkannan Eswaran, and Natarajan Sriraam. "Approximate entropy-based epileptic EEG detection using artificial neural networks." *IEEE Transactions on Information Technology in Biomedicine* 11.3 (2007): 288-295.
6. Yuan, Qi, et al. "Epileptic EEG classification based on extreme learning machine and nonlinear features." *Epilepsy research* 96.1-2 (2011): 29-38.
7. Subasi, Abdulhamit, and M. Ismail Gursoy. "EEG signal classification using PCA, ICA, LDA and support vector machines." *Expert systems with applications* 37.12 (2010): 8659-8666.
8. Acharya, U. Rajendra, et al. "Automated diagnosis of epileptic EEG using entropies." *Biomedical Signal Processing and Control* 7.4 (2012): 401-408.
9. Liang, Zhenhu, et al. "EEG entropy measures in anesthesia." *Frontiers in computational neuroscience* 9 (2015): 16.
10. Gosseries, Olivia, et al. "Automated EEG entropy measurements in coma, vegetative state/unresponsive wakefulness syndrome and minimally conscious state." *Functional neurology* 26.1 (2011): 25.

11. Thul, Alexander, et al. "EEG entropy measures indicate decrease of cortical information processing in Disorders of Consciousness." *Clinical Neurophysiology* 127.2 (2016): 1419-1427.
12. Rodríguez-Sotelo, Jose Luis, et al. "Automatic sleep stages classification using EEG entropy features and unsupervised pattern analysis techniques." *Entropy* 16.12 (2014): 6573-6589.
13. Huang, Kuan-Chih, et al. "An EEG-based fatigue detection and mitigation system." *International journal of neural systems* 26.04 (2016): 1650018.
14. Zhang, Xiaoliang, et al. "Design of a fatigue detection system for high-speed trains based on driver vigilance using a wireless wearable EEG." *Sensors* 17.3 (2017): 486.
15. The Theoretical and Applied Neuroscience Laboratory at the University of Victoria <https://www.krigolsonlab.com/>
16. Krigolson, Olave E., et al. "Choosing MUSE: Validation of a low-cost, portable EEG system for ERP research." *Frontiers in neuroscience* 11 (2017): 109.
17. Delgado-Bonal, Alfonso, and Alexander Marshak. "Approximate entropy and sample entropy: A comprehensive tutorial." *Entropy* 21.6 (2019): 541.
18. Rish, Irina. "An empirical study of the naive Bayes classifier." *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. No. 22. 2001.
19. Decision Trees: A simple way to visualize a decision <https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb>
20. Costa, Yandre MG, Luiz S. Oliveira, and Carlos N. Silla Jr. "An evaluation of convolutional neural networks for music classification using spectrograms." *Applied soft computing* 52 (2017): 28-38.

Authors

Muhammad Azam received his Bachelor degree in Electrical Engineering from the University of Victoria. Currently he is pursuing his Master's in Computer Science. His research interests include Artificial Intelligence, Computer Vision, and Computer Graphics.

Derek Jacoby obtained his PhD in Computer Science from the University of Victoria. He has more than 10 years of experience at Microsoft, leading teams in Windows system administration tools, and research on speech recognition. He founded Canada's first community biology lab at the Victoria Makerspace.

Yvonne Coady is a professor of computer science at the University of Victoria where she works with the MOD(ularity) Squad to improve the modularity of system infrastructure software. Her research interests include mixed reality systems, citizen science, advanced modularity across the software stack and distributed clouds, and new programming paradigms and pedagogy for immersive applications.

PROJECT OF SORTING SYSTEM FOR PLASTIC GARBAGE IN SORTING PLANT BASED ON ARTIFICIAL INTELLIGENCE

Janusz Bobulski and Mariusz Kubanek

Department of Computer Science,
Czestochowa University of Technology, Poland

ABSTRACT

An important element of a complex recycling process that is an integral part of municipal waste management is the sorting of materials that can be re-used. Manual sorting of garbage is a tedious and expensive process, which is why scientists create and study automated sorting techniques to improve the overall efficiency of the recycling process. An important aspect here is the preliminary division of waste into various groups, from which detailed segregation of materials will take place. One of the most important contemporary environmental problems is the recycling and utilization of plastic waste. The main problem under consideration in this article is the design of an automatic waste segregation system. A deep convoluted neural network will be used to classify images.

KEYWORDS

Convolutional Neural Network, Deep Learning, image processing, waste management, environmental protection, recycling.

1. INTRODUCTION

Plastics are synthetic organic matter obtained from e.g. natural gas, petroleum or cellulose as a result of the polymerization process. They are used for the production of packaging, clothes, some building and car components, toys and home appliances. It has been estimated that the total amount of manufactured plastic products in the world increased from 1.5 million tonnes (1950) to 245 million tonnes (data for 2008) [1]. The most important types of plastics include thermoplastics and thermosetting plastics. Thermoplastics can be subjected many times to heat treatment, during which they soften and melt, and after cooling again become hard. This property means that old products can be used for recycling. Examples are products made of PE (polyethylene), PS (polystyrene), PET or PVC (polyvinyl chloride). They constitute 80% of all plastics used. On the other hand, thermosetting plastics (duroplastics) can be melted and formed, but after forming they remain hard and do not soften under the influence of heating. These materials include: epoxides, phenoplast, polyurethanes, polytetrafluoroethylene (PTFE), polyester resins. These materials constitute 20% of all plastics used. Landfilling is undoubtedly the worst way to dispose of waste, both from an economic and ecological point of view. Research on plastics recycling has been conducted for years.

In order to facilitate recycling processes, the obligation to label the type of waste was introduced all over the world, and in the case of plastic waste - the type of plastic. Thanks to this, we know, for example, whether a given packaging can be recycled and what method. Plastic waste was divided into seven groups and marked with numbers from 1 to 7 in a triangle consisting of

David C. Wyld et al. (Eds): ICAITA, CDKP, SAI, NCO, CMC, SOFT, MLT, AdNLP - 2020

pp. 27-35, 2020. CS & IT - CSCP 2020

DOI: 10.5121/csit.2020.100903

arrows. They are: 1-PET - polyethylene terephthalate, 2-PE-HD - high-density polyethylene, 3 - PVC - polyvinyl chloride, 4 - PE-LD - low-density polyethylene, 5 - PP - polypropylene, 6 - PS - polystyrene, 7 - Other. Unfortunately, manual sorting of waste is expensive and laborious, and existing mechanical methods are inefficient. With this in mind, we decided to build a plastic waste sorting system using image processing methods and artificial intelligence. In this article, we present the results of using convolution neural network and deep learning techniques to recognize waste type based on the garbage image. There is a continuation of previous work that gave promising results in the classification into the four most common types of plastic in household waste. In this work, we present results for all seven groups [2]. The results allow the conclusion that the proposed system can be used in real conditions.

2. BACKGROUND

In many countries in the world waste segregation has already been introduced at the beginning of the recycling path, i.e. at home. Just people divide waste into groups such as plastic, metal, glass and organic/bio. The use of selectively automated techniques for these groups is easier than for municipal solid waste (MSW). Unfortunately, a large part of the waste is still collected in the form of the MSW, which is why the countries strive for the most effective reprocessing of waste materials. In order to do this, you should effectively sort the rubbish into individual factions and materials. Therefore, an important task is to isolate individual types of materials from the MSW. Therefore, techniques and procedures for segregating waste are used for the main groups of materials such as paper, glass, metal, wood, plastic and biomass by property system [3]. The biggest challenge, however, is the separation of various types of materials within a given group, i.e. sorting different colour of glass or different types of plastic. The problem of plastic garbage is interesting and at the same time important due to the possibility of recycling only some types of plastic (e.g. PET). To simplify the recycling process, international labelling of various types of plastics was introduced. These are:

- 1 - PET - polyethylene terephthalate,
- 2 - HDPE - high-density polyethylene,
- 3 - PVC - polyvinyl chloride,
- 4 - LDPE - low-density polyethylene,
- 5 - PP - polypropylene,
- 6 - PS - polystyrene,
- 7 - other.

The whole process of automatic sorting of materials suitable for reprocessing from MSW is complicated. There are many methods of waste sorting depend on type of material, that is: mechanical, electromagnetic, X-ray, grinding and the use of rotary equipment, manual, optic based and many others [3].

Commonly techniques often used physical features but ignored visual properties like colour, shapes, texture and size for the sorting of waste. In optical sorting, camera based sensors are used for the identification of waste fractions. In this section we present optical sorting techniques.

Sorting technique based on features like shape and colour was proposed by Huang et al. [4]. This method combines a 3D colour camera and laser beam over the conveyor belt. This technique formed triangles over the image from the camera on the base laser beam, so is called triangulation scanning. The technique achieves an accuracy of 99% for plastic fractions.

Spectral imaging is a combination of spectral reflectance measurement and image processing technologies. We may find several spectral imaging methods using NIR (near infrared), VIS (visual image spectroscopy) and HSI (hyperspectral imaging) [5].

A hyperspectral sensor produces images over a continuous range of narrow spectral bands and next system analysis the spectroscopic data. The conveyor system moves the waste fractions beneath the spectral camera acquires images. At the second stage data is pre-processing and reduction. Next to perform material classification special algorithm is applied. A set of compressed air nozzles is mounted at the end of the conveyor belt and depending upon the classifier decision, one of nozzles are triggered the waste into particular bins [6].

In the case of techniques based on spectroscopy, light is illuminated on plastic waste. Each type of plastic reflects a unique wave range, therefore NIR and laser sensors are used to read the wavelengths reflected from the material being tested. Then, based on the unique signature, it is classified by the processor unit.

Safavi et al. developed a technique that uses reflectometric spectroscopy to identify PP plastic in mixed wastes. The identification unit uses a spectrometer to analyse the reflected light from the sample and determine the type of material, and the compressed air nozzle ejects the elements to the appropriate boxes [7].

The HSI approach is used to classify high purity PP and PE plastics from mixed waste using near-infrared light NIR (1000-1700 nm) [8]. A typical spectroscopic system is equipped with a movable conveyor belt and a sensor system including a backlight and a NIR spectral camera. The image of the materials in the control zone is acquired by the NIR camera, and then it is processed by the classification algorithm.

In order to improve the efficiency of the classification algorithm, the principal components analysis (PCA) is used to reduce the data classification dimensions obtained from the spectral images [9]. Kassouf et al. [10] developed a quick way to classify plastics with the combination of MIR spectroscopy and independent component analysis (ICA). In addition, a more accurate classification is obtained by separating some plastic waste, e.g. LDPE and HDPE.

3. PROPOSED SYSTEM

After extracting plastic garbage from the MSW, a computer system based on image processing can be used to divide it into different types (Fig. 1). The method we propose uses an RGB digital camera and a computer with software for classifying plastic waste. In contrast, an air stream is used to direct the waste to a specific container. The software used in this system uses image processing techniques in the process of image pre-processing. However, convolution neural networks and deep learning [11] are used to recognize objects.

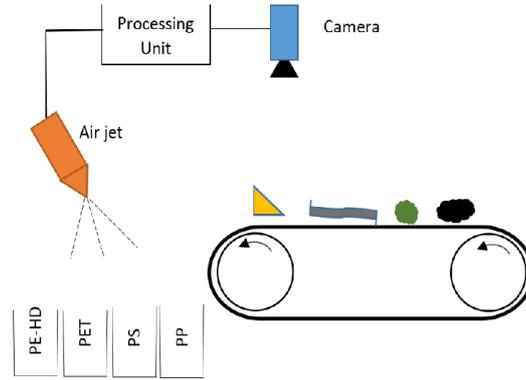


Fig. 1. Proposed system for plastic waste sorting.

3.1. Convolutional Neural Network

Convolutional neural network (CNN) is a feed-forward artificial neural network in which the organization of neurons is similar to the animal visual cortex. In order to recognize the shape of an object, the local arrangement of pixels is important. CNN starts with recognition of smaller local patterns on the image and concatenate them into more complex shapes. CNN was proved to be efficient especially in object recognition on an image. CNNs might be an effective solution to the waste sorting problem.

CNN explicitly assumes the input is an image and reflects it onto its architecture. CNN usually contains Convolutional layer, Pooling layer and Fully-connected layer. Convolutional layers and Pooling layers are stacked on each other, fully-connected layers at the top of the network outputs the class probabilities.

Convolutional layer

Convolutional layer consists of neurons connected to a small region of pixels (also called the receptive field) of previous layer. The neurons in same feature map share the same weights. Convolutional layer (CL) contains a set of learnable filters. One filter activates when a specific shape or blob of colour occurs within a local area [12]. Each CL has multiple filters F . Filter $f_i \in F$ is a set of learnable weights corresponding to the neurons in previous layer. Filter is small spatially (along width and height) and extends along the full depth c of the previous layer.

Pooling layer

Pooling layer (PL) is an effective way of non-linear down-sampling. It has as the convolutional layer receptive field and stride, however is not adding any learnable parameters. PL layer is usually put after the CL.

The receptive field r_n of neuron n in PL is 2 dimensional. It extends over a square of neurons in one feature map (or the neurons sharing the same third coordinate in case of not CL layer). Let us denote $width^{i-1}$ and h^{i-1} the spatial width and height of the previous layer. For each feature map there is P neurons in PL.

$$P = \frac{width^{i-1} \cdot h^{i-1}}{s_w \cdot s_h} \quad (1)$$

Max-pooling layer is the most frequently used pooling layer. Each neuron outputs the maximum of its receptive field. Usually the stride is the same as the size of receptive field. The receptive fields do not overlap, but touch. In most cases stride and size of receptive field are 2×2 .

The output (x_i, y_j, f_k) of the max-pooling neuron n with position (x_n, y_n, z_n) and receptive field r_n of size $f_w \times f_h$ in PL with stride (s_w, s_h) :

$$y(x_n, y_n, z_n) = \max_{0 \leq l < f_w} \max_{0 \leq m < f_h} v(x_n + l, y_n + m, f_k) \quad (2)$$

Inspired by a processes in the visual cortex of animals MAX-pooling layer amplifies the most present feature (pattern) of its receptive field and throws away rest. The intuition is that once a feature has been found, its rough location relative to other features is more important than its exact location. The pooling layer is effectively reducing the spatial size of the representation, does not add any new parameters – reducing them for latter layers, making the computation more feasible.

The idea of pooling layer was created back in the time with lack of computational power. Due to its destructiveness – throwing away 75% of input information in case of small 2×2 receptive field, the current trend prefers stacked convolutional layers eventually with stride and uses pooling layers very occasionally or discards them altogether [13].

CNN Structure

As a feed-forward artificial neural network, the CNN consists of neurons with learnable weights and biases. CNN's neurons still contains activation function and the whole network expresses single differentiable score function. The position of the pixel matters in comparison with MLP. It receives 3 dimensional space input (x, y, z) – the value of z -th channel of the pixel or occurrence of z -th feature of CL at position (x, y) . One pixel is usually made of three channels – red, green and blue.

The convolutional layer and pooling layer are locally connected to the outputs of the previous layer, recognizing or magnifying local patterns in the image. Pooling layer is usually put after the convolutional layer. This pair of layers is repeatedly stacked upon each other following with the fully connected layers at the top.

Usual architecture can be: input layer (IL), CL, PL, CL, PL, full-connected layer (FC), FC. Recent studies suggest stacking many CLs together with fewer PLs.

The fully connected layer is connected to all outputs of last pooling layer. The outputs of last pooling layer should already represent complex structures and shapes. The fully connected layer follows usually with another one or two layers finally outputting the class scores.

Back-propagation

The single evaluation is completely consistent with the feed-forward neural network. The input data or activations are passed to next layers, dot product is computed over which activation function is applied. Down-sampling the network using pooling layer might be present. At the end two or three fully connected layers are stacked. In order to use gradient descent learning algorithm, the gradient must be computed.

The usual back-propagation algorithm is applied with two technical updates. Classical back-propagation algorithm would calculate different partial derivatives of weights belonging to the neurons in same filter, however these must stay the same. Therefore, derivatives of loss function with respect to weights of neurons belonging to the same feature map are added up together.

The update of back-propagation itself is when dealing with max-pooling layers. The back propagating error is routed only to those neurons which have not been filtered with max-pooling. It is usual to track indices of kept neurons during forward propagation to speed up the back-propagation.

Autoencoder

Autoencoder is a feed-forward neural network where expected output is equal to the input of the network – its goal is to reconstruct its own inputs. Therefore, autoencoders are belonging to the group of unsupervised learning models [14]. Usually autoencoder consists of an input layer, one or many hidden layers and output layer. Since the idea of autoencoders is very similar to Restricted-Boltzman Machine, it is common for the structure of autoencoders to follow the rule: $|l_i| = |l_{(k-i)}|$. Let us denote the layer l_c , such that the number of neurons in l_c is lower than in any other layer. The l_c is an encoding layer. The feed-forward neural network consisting of layers l_i ; $i \leq c$, is called encoder. Expectedly, stacked layers l_j , such as $j \geq c$ is called decoder. Each autoencoder consists of encoder and decoder. The encoder can be used for compression. Unlike Principal Component Analysis analysis restricted to linear mapping, the encoder represents non-linear richer underlying structures of the data [15]. The activations of the l_c layer can be further used for classification. Fully-connected layers are appended with the size of the last corresponding to the number of labels. In our system, the autoencoder is used to encode the input signals.

Deep Autoencoder consists of many layers stacked on each other allowing to discover more complicated and non-linear structures of the data. Since it may be complicated to tune deep autoencoder network, commonly the training procedure is made of two steps:

Pre-training, each layer l_1, \dots, l_c is pre-trained. Firstly the pair l_0 as an example and l_1 as encoder is used. The goal is to find representation of l_0 in l_1 using the right optimizer. The weights l_0 to l_1 and l_1 to l_0 may be tied up representing Restricted Boltzman-Machine. When good representation of l_0 inputs is encoded in l_1 the pair l_1, l_2 is pre-trained further till pair l_{c-1}, l_c is reached.

Fine-tuning, the full network is *connected* and fine-tuned. In case of classification, the encodings of input data points can be used for classification training or the whole network $l_0 \rightarrow \dots \rightarrow l_c \rightarrow f_{c1} \rightarrow f_{c2}$ is part of the supervised learning.

4. EXPERIMENT

4.1. Structure of the Network

A number of important factors had to be taken into account when working on the appropriate selection of the network structure. First of all, the size of the input image was an important element. Too high resolution resulted in increasing the number of calculations, which resulted in fairly frequent overload of memory available computing unit. But, too low resolution of the input data could have prevented the achievement of the expected performance. Determined to conduct research for images with a resolution of 60 x 120 pixels. Another important element was the selection of the number and types of layers of the CNN network. Our network contained 16

layers. The first convolution layer consisted of 64 convolution filters with dimensions 9 x 9. Three layers of convolution encode information, transferred to a two-layer fully connected layer. The network diagram for 60 x 120 pixel images is shown in Table 1.

Table 1. Structure of the proposed network.

No	Name of layer	Parameters
1	Image input layer	60 x 120 x 3
2	Convolution Layer	64 filters, size 5 x 5
3	Max Pooling Layer	
4	ReLU Layer	
5	Cross Chanel Normalization Layer	
6	ReLU Layer	
7	Max Pooling Layer	
8	Convolution Layer	64 filters, size 5 x 5
9	ReLU Layer	
10	Max Pooling Layer	
11	Convolution Layer	64 filters, size 5 x 5
12	Fully connected layer	Inputs 4992, outputs 64
13	ReLU Layer	
14	Fully connected layer	Inputs 64, outputs 7
15	ReLU Layer	
16	Classification layer	7

4.2. Input Data

Preparation of input data for the learning and testing phase was important in the context of correct classification of objects in natural working conditions. In the case of deep neural networks, as many data as possible should be collected for each identified class. In our case, it was necessary to collect photos of classified waste. We adopted a simplified model where there could be only one waste within the camera lens. This approach does not reflect the natural working conditions, but for research needs it gives sufficient opportunities to generate a properly functioning network. All collected images represented objects classified into seven considered classes: PET, PE-HD, PVC, PE-LD, PP, PS and Other. These images came from the commercial Garbage database[16], and their samples maybe seen on Fig. 2. To increase the number of images in individual classes, we have modified existing images by flipping and rotating. Images from all classes were rotated by angle 18 degrees, In this way, we obtained over 200 000 images. We chose randomly for teaching of 10,000 per class and for testing of 1,000 for each class.

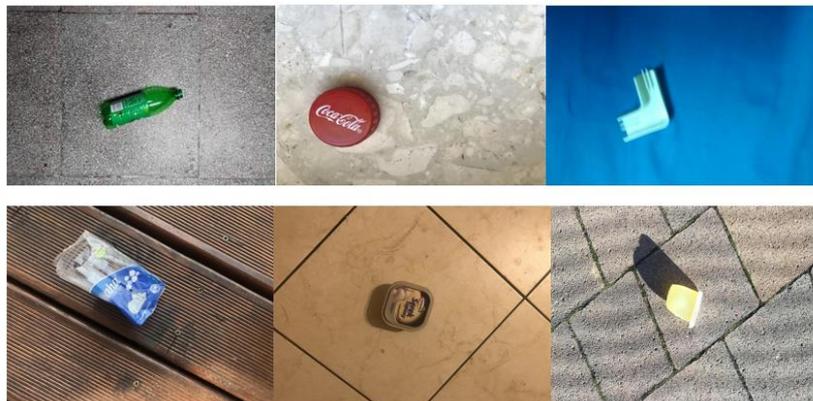


Fig. 2. Samples images of plastic waste

Table 1. Learning results of a 16-layer network (image resolution 60 x 120 pixels)

No.	Initial learn rate	Learned rate per period	Max epoch	Regularization	Accuracy [%]
1	0.0001	10	20	0.1	85
2	0.001	5	20	0.1	90
3	0.0001	5	20	0.1	88
4	0.0001	15	20	0.1	86
5	0.001	10	30	0.1	91
6	0.01	10	30	0.1	34
7	0.001	10	30	0.1	85
8	0.001	10	30	0.1	89
9	0.001	15	30	0.01	88
10	0.001	15	30	0.01	87

Table 2 presents learning stages conducted for our network using images with a resolution of 60 x 120 pixels. Analysing the obtained results, it we see our network reach good results for fifth stage, when it achieved 91%. Ten epochs were sufficient to obtain an acceptable level of accuracy. Further learning, even with a reduced learning rate, no longer significantly affects accuracy. Average accuracy of 91% is a very good result as a fairly small number of iterations. Regarding the other learning parameters, the best results were obtained for Initial learn rate 0.01, Learned rate per period 10, Max epoch 30 and Regularization 0.1. The last one parameter didn't make strong influence on accuracy. The biggest impact on the change of accuracy was the change Initial learn rate to 0.1.

We used five sets of images, were each class contained 1000 images, and we obtain the highest average result 91% for parameters presented at line 5 in Table 3. Accuracy at presented level is acceptable for the proper functioning of the system in real conditions.

5. CONCLUSION

The conducted research has shown that the 15-layer network proposed by us allows achieving high efficiency for images with a resolution 60 x120. Classification of segregated waste into seven main classes takes place in most cases without error. Of course, this is to a certain extent caused by the artificially increased number of individual class representatives. Further work will mainly consist of extending the database of segregated waste images with photos of waste in more realistic conditions. Hence, efforts to obtain recordings of waste on a conveyor belt from enterprises dealing with waste segregation. Our research in the future will assume the possibility of training the network while working in real conditions, which is possible to implement with our proposal. After introducing modifications to the training database, we also want to determine the accuracy for real images of waste taken from the conveyor belt during the segregation process.

6. ACKNOWLEDGEMENTS

Supported by the project financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 – 2022. Project number 020/RID/2018/19 the amount of financing 12,000,000 PLN.

REFERENCES

1. GUS homepage, <https://stat.gov.pl/>, last accessed 2019/09/25.
2. Bobulski, J., Kubanek, M.: Waste Classification System Using Image Processing and Convolutional Neural Networks. In: Rojas, I., Joya, G. (eds.), *Advances in Computational Intelligence*, pp. 350–361 (2019).
3. Gundupalli, S.P., Hait, S., Thakur A., A review on automated sorting of source-separated municipal solid waste for recycling. *Waste Management* 60, pp. 56–74 (2017).
4. Huang, J., Pretz, T., Bian, Z.: Intelligent solid waste processing using optical sensor based sorting technology. 3rd International Congress on Image and Signal Processing (CISP), vol. 4. IEEE, pp. 1657–1661, (2010).
5. Pieber, S., Meirhofer, M., Ragossnig, A., Brooks, L., Pomberger, R., Curtis, A.: Advanced Waste-Splitting by Sensor based Sorting on the Example of the MTPlantOberlaa. *Tagungsband zur 10. DepoTech Conference*, pp. 695–698, (2010).
6. Tatzert, P., Wolf, M., Panner, T.: Industrial application for inline material sorting using hyperspectral imaging in the NIR range. *Real-Time Imag.* 11 (2), pp. 99-107, (2005).
7. Safavi, S.M., Masoumi, H., Mirian, S.S., Tabrizchi, M.: Sorting of polypropylene resins by color in MSW using visible reflectance spectroscopy. *Waste Manage.* 30 (11), pp. 2216-2222, (2010).
8. Serranti, S., Gargiulo, A., Bonifazi, G.: Characterization of post-consumer polyolefin wastes by hyperspectral imaging for quality control in recycling processes. *Waste Manage.* 31 (11), pp. 2217-2227, (2011).
9. Serranti, S., Gargiulo, A., Bonifazi, G.: Classification of polyolefins from building and construction waste using NIR hyperspectral imaging system. *Resour. Conserv. Recycl.* 61, pp. 52-58, (2012).
10. Kassouf, A., Maalouly, J., Rutledge, D.N., Chebib, H., Ducruet, V.: Rapid discrimination of plastic packaging materials using MIR spectroscopy coupled with independent components analysis (ICA). *Waste Manage.* 34 (11), pp. 2131-2138, (2014).
11. Wang, M., Wang, Z., Li, J.: Deep Convolutional neural network applies to face recognition in small and medium databases. 4th International Conference on Systems and Informatics, ICSAI 2017 Volume 2018-January, 4 January 2018, pp. 1368-1372, (2018).
12. Convolutional Neural Networks (CNNs / ConvNets). 2016. url: <http://cs231n.github.io/convolutional-networks/>, last accesses on 09/01/2019.
13. Goodfellow I., Benigo Y., Courville A.: *Deep Learning*, The MIT Press, 800 (2016)
14. Kaiming He.: *Deep Residual Learning for Image Recognition*, url: <https://arxiv.org/abs/1512.03385> last accesses on 12/15/2018.
15. Hinton G.E., R. R. Salakhutdinov R.R., *Reducing the Dimensionality of Data with Neural Networks*, url: <http://science.sciencemag.org/content/313/5786/504>, last accesses on 15/07/2019.
16. Garbage Image Database, url: <http://garbage.eu5.com> last accesses on 01/09/2019.

EVALUATING AND VALIDATING CLUSTER RESULTS

Anupriya Vysala and Dr. Joseph Gomes

Department of Computer Science, Bowie State University, USA

ABSTRACT

Clustering is the technique to partition data according to their characteristics. Data that are similar in nature belong to the same cluster [1]. There are two types of evaluation methods to evaluate clustering quality. One is an external evaluation where the truth labels in the data sets are known in advance and the other is internal evaluation in which the evaluation is done with data set itself without true labels. In this paper, both external evaluation and internal evaluation are performed on the cluster results of the IRIS dataset. In the case of external evaluation Homogeneity, Correctness and V-measure scores are calculated for the dataset. For internal performance measures, the Silhouette Index and Sum of Square Errors are used. These internal performance measures along with the dendrogram (graphical tool from hierarchical Clustering) are used first to validate the number of clusters. Finally, as a statistical tool, we used the frequency distribution method to compare and provide a visual representation of the distribution of observations within a clustering result and the original data.

KEYWORDS

Hierarchical Agglomerative Clustering, K-means Clustering, Internal Evaluation, External Evaluation, Silhouette

1. INTRODUCTION

Clustering is grouping sets of data objects into multiple groups or clusters so that objects within the cluster have high similarity, but are very dissimilar to the objects in the other clusters [2]. The clustering algorithms can be used on both normalized and non-normalized data. If users have normalized data, the number of iteration of the algorithms will be fewer. Therefore, in most of the situation, normalized data offers good outcome as compared to non-normalized data[3]. In machine learning, there are both supervised and unsupervised learning algorithms. In the case of supervised learning, we feed training data along with predefined labels, whereas in the case of unsupervised learning we only feed features when we train data as the true label is not known. In general, unsupervised algorithms make inferences from datasets using only input vectors without referring to the labeled outcomes. K-means is one of the Unsupervised learning algorithms. There are many types of clustering such as exclusive, overlapping, hierarchical, etc. In the case of exclusive clustering, each sample in the dataset will belong to one of the clusters, whereas in the case of overlapping clusters there can be data points that are in more than one cluster. In the case of hierarchical clustering, a tree-like structure is produced.

In this paper, we first validate the number of clusters by various internal evaluation techniques and then compare those results with the K-means elbow method and hierarchical agglomerative clustering. Later we do the cluster analysis based on internal evaluation techniques and frequency distribution method for both K-means and Hierarchical Clustering Algorithm. We have implemented these in Python.

The rest of the paper is organized as follows: Section 2 details about K-means and Hierarchical Clustering, Section 3 offers surveys of related works. Section 4 offers various external and internal evaluation methods. Section 5 explains the proposed methodology. Section 6 discusses the findings from these results. Finally, we conclude by briefly explaining our contributions and further works.

2. K-MEANS AND HIERARCHICAL CLUSTERING

2.1. K-means

K-means is one of the popular unsupervised machine learning algorithms for clustering. K-means is fast, compared to other clustering algorithms as it doesn't require calculating all of the distances between each observation and every other observation. K-means can be used for clustering very large datasets. But traditional k means algorithm does not always generate good quality results as automatic initialization of centroids affects final clusters as mentioned in [4].

In this paper, we used the k-means algorithm on a normalized dataset in order to generate quality clusters, it minimized the k-means inertia criteria (within-cluster sum of square inertia) and gave high score in case of various external evaluation techniques, which can be seen in section 6, Table 1.

The k-means model was implemented using the scikit-learn library in python where we used the k-means++ for initialization as 'k-means++' selects initial cluster centers for k-means clustering in a smart way to speed up convergence. In our experiment, the training data was split into random train and test subsets using 80-20 rule and the random state was set to an integer value to have a reproducible result for documenting, in order to avoid different solutions every time, we do sampling.

2.1.1. Algorithm (K-means)^[2]

The algorithm proceeds as follows:

1. Randomly initialize 'k' clusters centroids
2. Assign each data points to its closest cluster centroid
3. Compute the centroid of the new partition formed by taking the average of points assigned to that cluster
4. Repeat steps 2, and 3 until convergence is obtained

2.2. Hierarchical Clustering

Another type of clustering technique is called Hierarchical Clustering. The two types of Hierarchical clustering are Agglomerative clustering and Divisive clustering. In the case of hierarchical agglomerative clustering it begins by putting each observation into its own separate cluster and then it builds new higher-level clusters by grouping clusters based on distance. The graphical tool to get insight into the clustering solution is called a dendrogram.

In hierarchical clustering, as a first step, find the distance matrix and then examine all the distances between all the observations and pairs together the two closest ones to form a new cluster. So finding the first cluster simply means looking for the lowest number in the distance matrix and merging the two observations that the distance corresponds to into a new cluster. Hence there will be one fewer cluster than the number of observations. To determine which

observations will form the next cluster certain linkage methods are used. The different linkage methods used are single linkage, complete linkage, average linkage, and ward.

The Single linkage method defines cluster proximity as the shortest distance (min) between two points, x , and y , that are in different clusters, A and B . Complete linkage defines cluster proximity as the farthest(max) distance between two points, x and y , that are in different clusters, A and B . Average proximity defines cluster proximity as the average distance between two points, x and y , that are in different clusters, A and B . In ward's method, the approach of calculating the similarity between two clusters is exactly the same as group average except that Ward's method calculates the sum of the square of the distances 'Pi' and 'Pj' where 'Pi' is a data point in cluster 1 and 'Pj' is a data point in cluster 2. In our paper, we have compared the results using all these methods for the IRIS dataset and figured out which one gives a better score.

2.2.1. Algorithm (Hierarchical Clustering) ^[5]

1. Start with n clusters where each data object forms a single cluster
2. Repeat step 2 until there is only one cluster left
 - (a) Find the closest pair of clusters
 - (b) Merge them
3. Return the tree of cluster-merger

3. RELATED WORK

The paper [1] does a comparison of K-Means Clustering and CLARA Clustering on Iris Dataset, which uses Euclidean distance and Manhattan Distance as a dissimilarity measure, respectively. It is an extension of the K-Medoids Clustering algorithm, which uses a sampling approach to handle large datasets. The paper proves that CLARA Clustering using Manhattan distance is better than K-Means Clustering with Euclidean distance.

Dr. Manju Kaushik et al. [2] presented a comparative study of K-means and Hierarchical Clustering techniques in terms of Clustering Criteria, performance, size of a dataset, sensitivity to noise, quality, execution time, number of clusters, etc.

In [4] Deepali Virmani et al., have first preprocessed the dataset based on the normalization technique and then generated effective clusters. They also assigned a weight to each attribute value to ensure the fair distribution of clusters. Their algorithm has proved to be better than the traditional K-means algorithm in terms of execution time and speed.

Weka data mining tool version 3.7 was used in [5] for testing accuracy and running time of k-means and hierarchical clustering algorithms on the IRIS and Diabetes dataset. In this paper, they concluded that the accuracy of k-means for the IRIS dataset having 'real' datatype attributes was greater compared to hierarchical clustering and for the diabetes dataset having both integer and real attributes accuracy of hierarchal clustering was greater than the k-means clustering algorithm. In the case of running time, k-means was faster.

Rousseeuw [6] proposed a technique called silhouette analysis which is based on the comparison of object tightness and separation. This tool can be used to validate the number of clusters in the case of the clustering algorithm.

Kalpit et al., [7] compared K-means and K-medoids algorithms using the Iris dataset. The results showed that K-medoids is better than K-means at scalability for a larger dataset. K-medoids showed its superiority over k means in execution time and sensitivity towards outliers.

In [9], Tippaya Thinsungnoena et al. studied two interesting sub-activity of the clustering process -selecting the number of clusters and analyzing the result of data clustering using SSE and silhouette analysis.

In [11] Huda et al. studied the several applications of the k-means algorithm in data mining and pattern recognition. They showed that k-means produces a promising result in both as a clustering method in data mining and for segmenting images in pattern recognition. Thus the paper concluded that k-means clustering is an efficient algorithm in both areas.

T. Velmurugan et.al., from the experiments they performed, concluded in [13] that K-Medoids is more robust than K-Means clustering in terms of noise and outliers, although K-Medoids is good for only small datasets.

4. VARIOUS EXTERNAL AND INTERNAL EVALUATION METHODS

4.1. External Evaluation

Homogeneity: If each cluster in the clustering result contains only data points that are members of a single class, then it satisfies homogeneity.

Completeness: If all the data points that are members of a given class are elements of the same cluster in the clustering result, then it satisfies completeness.

V- measure: The V-Measure is defined as the harmonic mean of homogeneity and completeness of the clustering.

4.2. Internal Evaluation

4.2.1. Silhouette Analysis

According to Rousseeuw [6], Silhouette is a tool used to assess the validity of clustering. Silhouette analysis helps to find the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters. The Silhouette score has a range of [-1, 1]. Thus, if the Silhouette score has a value near +1 that indicates the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.

4.2.2. Algorithm (Silhouette Score for each Data Object)^[6]

- (i) Find the mean distance of the point (i) with respect to all other points in the cluster it is assigned (say a). Let it be named as a(i).
- (ii) Find the mean distance of the point (i) with respect to its closest neighboring cluster (say b). Let it be named as b(i).
- (iii) Silhouette score for particular data point,

$$S[i] = [b(i)-a(i)] / \max[b(i),a(i)]$$

For the Silhouette score, $S[i]$ to be close to 1, $a(i)$ has to be very small compared to $b(i)$, that is $a(i) \ll b(i)$. The average Silhouette score can be calculated by taking the mean of silhouette scores of all samples in the dataset. The silhouette score ranges from -1 to $+1$, larger values represent that the data object is very similar to its own cluster and dissimilar to neighboring clusters. These metrics are independent of the absolute values of the labels; thus a permutation of the class or cluster label values will not change the score values in any way.

4.2.3. Within Sum of Squares (WSS)

The WSS (Within Sum of Squares) is used to estimate the number of clusters. If k is the number of clusters then as 'k' increases, the sum of squared distance tends to zero. i.e., if we set 'k' to its maximum value n (where n is the number of samples) then each sample will form its own cluster thus the sum of squared distances will be equal to zero. In this paper, we use the K-means inertia plot to validate the number of clusters based on the distortion/ WSS.

WSS measures the squared average distance of all the points within a cluster to the cluster centroid. To calculate WSS, first find the Euclidean distance between a data point and the centroid to which it is assigned. Then iterate this process for all data points in that cluster, and then sum the values for the cluster and divide by the number of data points. Finally, you calculate the average across all clusters to get the average WCSS.

5. PROPOSED METHODOLOGY

For a given labeled dataset, perform the feature scaling and figure out which one gives better scores in terms of WSS. Perform clustering using those normalized datasets and calculate the score using external and internal evaluation methods. Then select the one that has high scores for all the metrics. Finally, cluster analysis is performed by means of the frequency distribution. A frequency distribution is a tabular representation of data showing the outcomes in named classes. Thus this analysis helps in comparing the clustered result with the actual classes in the dataset. This whole process is then repeated for hierarchical agglomerative clustering as well. The graphical tool (dendrogram) in the case of hierarchical clustering helped to get insight into the number of cluster solution and that was compared with the k-means inertia plot to have a double validation on the number of clusters chosen. The proposed methodology helps in validating the number of clusters and also in evaluating clustering results.

6. EXPERIMENTAL SETUP AND RESULTS

For the experiment, we used the famous IRIS dataset. Python programming was used for both the clustering techniques. Initially, the dataset was split into 80-20 for training purposes. The number of records given for training the model was 120 samples with 4 features. As the IRIS dataset is labeled, the number of unique values in the training data after sampling, but before clustering is given below

```

Unique Values : ['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
Occurrence Count : [43 38 39]
Unique Values along with Count in the training data
Iris-setosa Occurs : 43 times
Iris-versicolor Occurs : 38 times
Iris-virginica Occurs : 39 times

```

Figure 1: Unique Values in the Training data

Table 1: Scores obtained based on K-means after performing various feature scaling on the dataset.

K-means				
	Inertia	Homogeneity	Completeness	V-measure
Before Scaling	65.17	-	-	-
Applying Standardization	106.5	-	-	-
Applying PCA	53.89	-	-	-
Applying MaxAbs	21.58	0.94	0.94	0.94
Applying MinMax	54	-	-	-
Normalizer	0.24	0.93	0.94	0.94

The K-means clustering was performed on normalized and non-normalized IRIS datasets using various methods such as standardization, minmax, MaxAbs, normalizer and also by dimensionality reduction. K-means, when applied after normalizing the IRIS dataset gave the best score in terms of Homogeneity, V-measure, and Completeness.

The homogeneity, V-measure, and completeness were calculated only for those 2 cases (MaxAbs and Normalizer) where the distortion is minimum and out of those, the normalized dataset was chosen for further analysis. 'Normalization based K-means Clustering algorithm [4]', has proved that applying normalization before performing clustering will improve the execution time and speed.

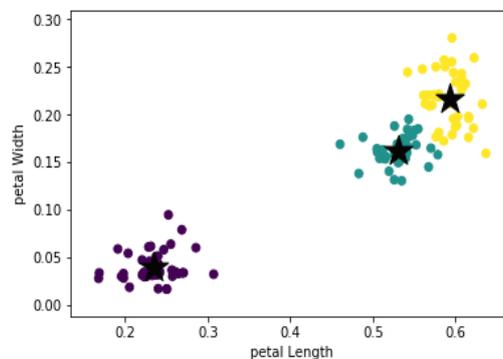


Figure 2: Scatter plot after applying K-means on normalized data.

Species	Iris-setosa	Iris-versicolor	Iris-virginica
labels			
0	43	0	0
1	0	36	0
2	0	2	39

Figure 3: Frequency distribution of labels after K-means clustering on training data

Eventhough for labeled data we know the number of classes in prior, we have compared the result of elbow method with that of a dendrogram using 'ward' as the linkage method for a

second level of validation on the number of clusters, which can be used as a method in the case of unlabeled data for future work.

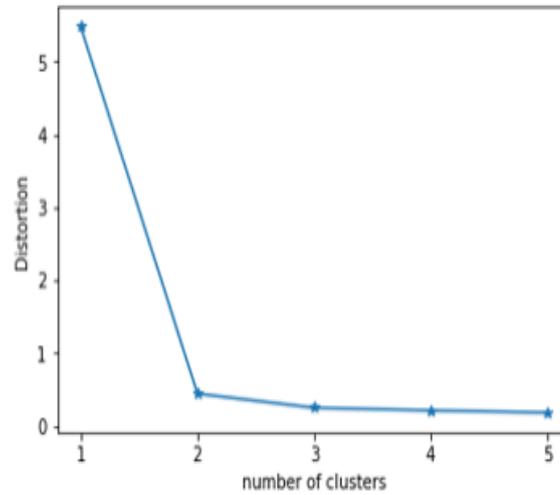


Figure 4: K-means inertia plot.

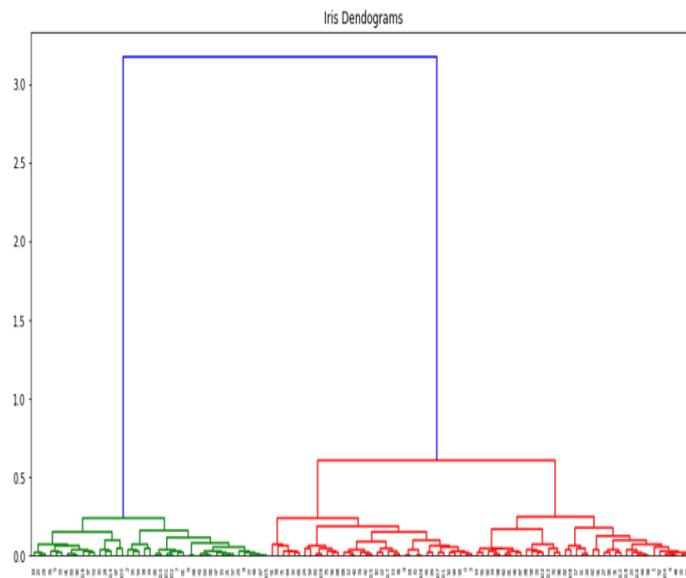


Figure 5: Iris Dendrogram.

Using the dendrogram we have taken the cut off at distance 0.5, as the jump in the distance is pretty obvious. If we draw a horizontal line at 0.5 then we end up with three clusters. On the x-axis, it provides you with the details on which clusters get merged at various distances. Thus comparing both the elbow and dendrogram we can conclude that the number of clusters for this dataset is 3 which is true based on the labels. In the case of Hierarchical clustering on the IRIS dataset, we tried different linkage methods and the highest score was obtained when the linkage method used was ‘average linkage’.

Comparing different hierarchical linkage methods on the IRIS dataset with cluster count as three produced the following results.

Table:2 Hierarchical Clustering on Normalized IRIS dataset.

Hierarchical Clustering on Normalized IRIS dataset			
Linkage methods Used :	Homogeneity	Completeness	V-measure
Ward	0.88	0.885	0.884
Average	0.93	0.93	0.93
Complete	0.90	0.91	0.90
Single	0.59	0.94	0.73

Species labels	Iris-setosa	Iris-versicolor	Iris-virginica
0	0	1	38
1	43	0	0
2	0	37	1

Figure 6: Frequency distribution table after applying average as the linkage method

No of Clusters	Average Silhouette Score
2	0.67
3	0.55
4	0.50
5	0.49
6	0.39

Table 3: Average Silhouette score based on k-means when cluster count =2,3,4,5,6.

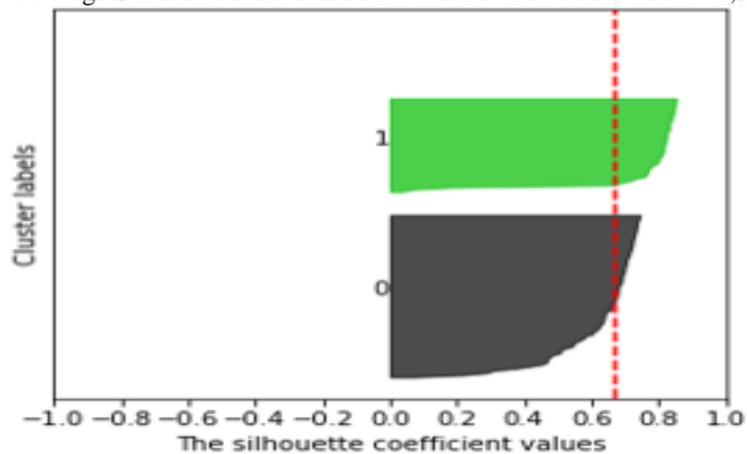


Figure 7: Silhouette plot for K-means clustering on the IRIS dataset when cluster count is 2.

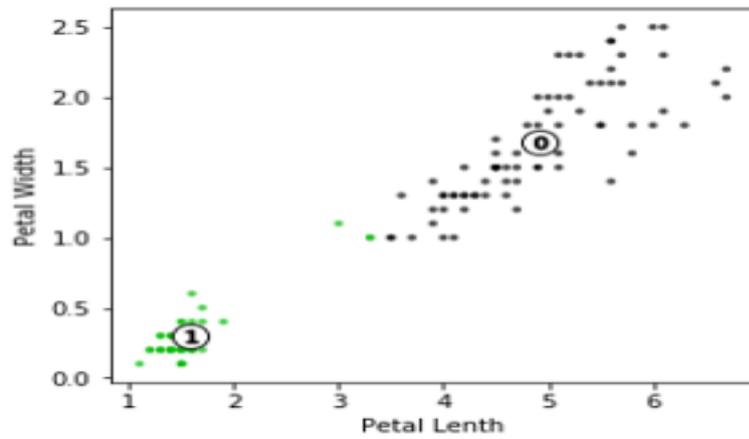


Figure 8: Visualizing the clustered IRIS data when cluster count is 2.

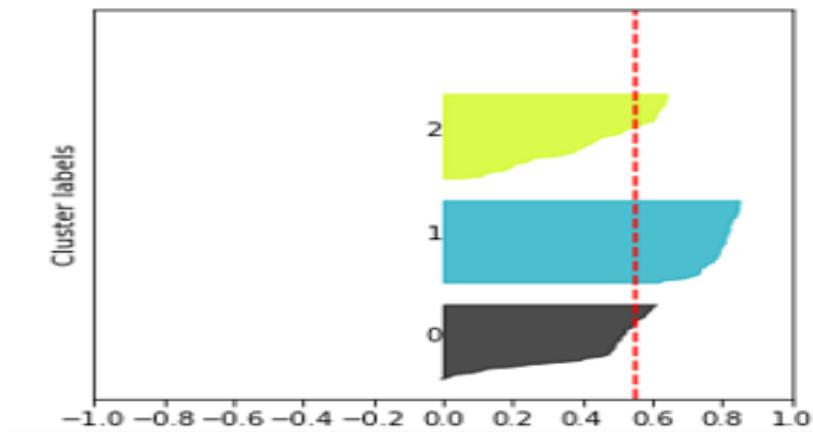


Figure 9: Silhouette plot for K-means clustering on the IRIS dataset when cluster count is 3.

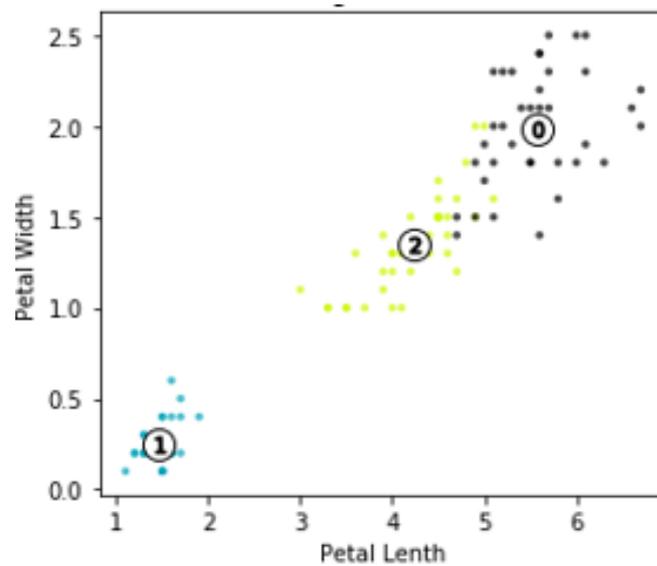


Figure 10: Visualizing the clustered IRIS data when cluster count is 3.

The figure (7,9) shows the silhouette plot for when cluster count is 2 and 3 and the figure (8,10) shows the clustered data points when cluster count is 2 and 3. Even though the average silhouette score is highest when cluster count is 2, we choose the number of clusters as 3 since the silhouette plot for cluster 0 when cluster count is 2 is bigger in size owing to the grouping of the 2 sub-clusters into one big cluster. However, when cluster count is 3, all the plots are more or less of similar thickness and hence are of almost similar sizes as can be verified from the labeled scatter plot. Also when cluster count is 3, the plot for each cluster is above the mean value (0.55) represented by a dotted line, and the width of the plot is uniform as possible. This shows that on average each datapoint is more similar to the points on its own cluster. Thus we can conclude that the number of clusters for this IRIS dataset has to be 3 to get better cluster results in terms of accuracy while performing K-means and Hierarchical Clustering.

7. CONCLUSION

In general, in this paper, we cross-checked the result of the dendrogram from the hierarchical agglomerative clustering and elbow method in K-means clustering to validate the best number of clusters as in real world most of the datasets come without true labels. Then to evaluate the quality of clustering based on different feature scaling and setting the number of clusters as 3, we calculated the V-measure, completeness and homogeneity score. These external validation methods were used for evaluating the quality of clustering. As a final step, we quantify the correctness of the partition by comparing the frequency distribution in the classes in the original training data with the clustered data. As an internal evaluation method, Silhouette analysis and WSS were also done on the training data. All the results are consistent in the sense that Silhouette and WSS present an appropriate number of clusters at the same k-value with regard to maximum average silhouette value and knee point in the case of WSS for the IRIS dataset. Also, we proved from the frequency distribution table that using k-means++ as the initialization procedure for the k-means, improved the clustering results using a clever seeding of the initial cluster centroids. The frequency table shows that the quality of the clustered classes was better while using k-means++ for cluster initialization than the result obtained while using K-medoids in paper [7] for the same IRIS dataset.

8. FUTURE WORK

As future work, we will use other labeled datasets to validate our methodology. Those datasets will be of different sizes to make sure that our method works well on all different datasets.

REFERENCES

- [1] Tanvi Gupta, Supriya P. Panda, (2018) "A Comparison of K-Means Clustering Algorithm and CLARA Clustering Algorithm on Iris Dataset", International Journal of Engineering & Technology.
- [2] Dr. Manju Kaushik, Mrs. Bhawana Mathur, (2014) "Comparative Study of K-Means and Hierarchical Clustering Techniques", International Journal of Software and Hardware Research in Engineering, Vol 2.
- [3] KM Archana Patel, Prateek Thakral, (2016) "The Best Clustering Algorithms in Data Mining", International Conference on Communication and Signal Processing.
- [4] Deepali Virmani, Shweta Taneja, Geetika Malhotra, "Normalization based K-means Clustering Algorithm".
- [5] Nidhi Singh, Divakar Singh, (2012) "Performance Evaluation of K-Means and Hierarchical Clustering in Terms of Accuracy and Running Time", International Journal of Computer Science and Information Technologies, Vol. 3.
- [6] Rousseeuw, P. J. (1987) "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics, Vol. 20, pp.53-65.

- [7] Kalpit G. Soni, Dr. Atul Patel, “Comparative Analysis of K-means and K-medoids Algorithm on IRIS Data”, International Journal of Computational Intelligence Research ISSN 0973-1873, Vol.13.
- [8] UCI Repository of Machine Learning Databases. Available: archive.ics.uci.edu/ml/
- [9] Tippaya Thinsungnoena, Nuntawut Kaoungku, Pongsakorn Durongdumronchai, Kittisak Kerdprasop, Nittaya Kerdprasop, (2015) “The Clustering Validity with Silhouette and Sum of Squared Errors”, Proceedings of the 3rd International Conference on Industrial Application Engineering.
- [10] Han, Jian, and Kamber (2006) Data mining concepts and techniques (2nd ed.), United States of America: Morgan Kaufman Publishers.
- [11] Huda Hamdan Ali1, Lubna Emad Kadhum2, “K- Means Clustering Algorithm Applications in Data Mining and Pattern Recognition”, International Journal of Science and Research (IJSR)
- [12] Aggarwal, C. C., Reddy, C. K. (2013) Data Clustering: Algorithms and Applications, Vol.31, CRC Press, Hoboken, New Jersey, p.648. (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, ISBN: 1466558210)
- [13] T.Velmurugan, T. Santhanum, (2008) “Performance Analysis of K-Means and K-Medoids Clustering Algorithms for A Randomly Generated Data Set”, International Conference on Systemics, Cybernetics and Informatics, pp.578-583.

AUTHORS

Anupriya Vysala is a Graduate Student in the Department of Computer Science at Bowie State University. She has her MS degree in Computer Science from George Washington University.



Dr. Joseph Gomes is an Associate Professor in the Department of Computer Science at Bowie State University. He has his MS and DSc degree in Computer Science from George Washington University.



DATA-DRIVEN TECHNIQUES FOR MUSIC GENRE RECOGNITION

Sergio Santiago Rentería, Jesus Leopoldo Llano and
Francisco Javier Cantú-Ortiz

Tecnológico de Monterrey, México

ABSTRACT

After the digital revolution, it is not strange to see data science taking interest in music. The sheer amount of available content opens a plethora of possibilities for studying music and its social impact from a data analytic perspective. This paper studies the relationship that exists between, song features and their corresponding genre, to provide data-mining tools for music recommendation and sub-genre identification. For the first task, we compared different classification models, including Random Forests, Fully-connected neural networks and Logistic Regression. For the latter, we carried out cluster analysis and dimensionality reduction for data visualisation. Overall, Random Forest models had better performance in genre classification than Fully-connected networks, but they suffered from overfitting. Moreover, the highest accuracy obtained was too low (64%) to be of use for genre recognition applications. Nevertheless, we think our results show the limitations of hand-crafted features and point towards more sophisticated deep learning techniques.

KEYWORDS

Music Information Retrieval, Data Mining, Automated Music Recommendation, Classification

1. INTRODUCTION

One of the most important skills that humans possess is the capacity of pattern recognition. This skill has allowed the emergence of systematic studies of nature such as physics and drives innovation and development in many other fields ranging from arts to engineering. Out of all the patterns that surround us, music stands out as a universal activity with multiple forms and complex manifestations across cultures. There is increasing evidence that all humans, not just highly trained individuals, share a predisposition for music in the form of musicality, which is defined as a spontaneous developing set of traits based on and constrained by our cognitive abilities and their underlying biology [4].

On the other side, the emergence of digital technologies has dramatically shifted how music is produced, distributed and consumed. It has made possible data-driven studies on the impact of music on human behaviour and the many ways in which we enjoy it. Nowadays, more information about artists and songs is available, new music is capable of reaching more people further and faster than ever before, and recommendations are easily tailored automatically based on user behaviour.

The area of Music Information Retrieval (MIR) arises as a way to tackle the challenges associated with effectively interacting and accessing increasingly large collections of music and all the associated data such as styles, genres, artists, lyrics, and music reviews. Algorithms

developed for these purposes employ sophisticated digital signal processing and machine learning techniques to extract musical features from audio signals and metadata [7]. Other relevant applications of this field include audio identification, score following, digital musical instruments and audio compression algorithms [7].

When listening to a song most people will not identify all of their features, like the tonality or the metre, but the overarching patterns that appear in songs allow individuals to recognise musical genres. Moreover, there are certain patterns and structures in music that people particularly enjoy, meaning that the most popular (hit) songs are those with patterns that better resonate with the listener. However, we should have in mind that what makes us like a song depends also on extra-musical factors such as the exposure effect and the social context in which music is consumed. Hit song science attempts the herculean task of predicting music success by finding features correlated with music popularity [7].

Understanding music from a data-driven perspective has become vital for the music industry. Techniques for identifying the contrasting and common characteristics in musical genres, extracting the similarities between music that become popular in contrast to music that doesn't could be used to enhance music recommendation and music production. Within these lines, the present article is focused on the application of data mining techniques to music genre recognition and sub-genre identification.

Our main interest is to explore the applications of data science in music recommendation systems and to study how they can improve streaming services such as Spotify and Apple Music. We structure this paper using the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework [10] to provide an implementation of this methodology in the field of Music Information Retrieval.

2. METHODOLOGY

To study common and contrasting characteristics of different music genres, we employ a series of data mining algorithms to extract patterns from the GTZAN genre collection [11]. This is an audio database containing a total of 1000 audio recordings across ten musical genres with 100 instances of 30 seconds each [11]. Below we give an overview of data mining techniques that have been used in this problem.

Different approaches to song recommendation and genre classification have been tried before including traditional algorithms and deep neural networks. Aaron et al. trained a Convolutional Neural Network to predict latent factors from music audio [12]. Gwardys and Grzywczak used transfer learning to leverage features from a model trained on ImageNet dataset in a genre recognition model based on MFCC spectrograms. They obtained an accuracy of 78% in GTZAN database by using SVMs over the CNN representations [3].

More recently, Yang and Zhang used a map of eight musical features as inputs of a CNN, their best model obtained 91 % accuracy on the GTZAN database [15]. Banitalebi-Dehkordi proposed a music genre classification method using Fast Fourier Transform (FFT) to extract short-term features from segments of each spectrogram. They obtained 95.7% Accuracy over GTZAN database [2].

Besides these approaches, there are other techniques and generic data mining algorithms that can be adapted to tackle the problem of music genre recognition. The most common approach is the use of statistical classification methods. An example of this can be seen in a paper by Schindler and Rauber [17] where the authors studied the effects of using a series of features extracted using

the Echonest Analyser [18]. They evaluated this approach in the task of music genre classification with commonly used classifiers in the MIR field, such as KNN, SVM, Random Forest, Naive Bayes and J48 Decision Tree. The authors applied these methods to a set of 4 databases with different subsets of extracted features. They obtained at most 66.9% accuracy in the GTZAN music genre recognition task.

Another example of the application of data mining algorithms adapted and applied to MIR can be seen in a paper by Kotsifakos et al. They carry out genre classification by combining the k-Nearest Neighbours classifier with Subsequence Matching with Bounded Gaps and Tolerances (SMBGT). SMBGT is used to extract similarity features between pairs of songs that characterise them. While k-NN using the extracted features assigns a genre to each song based on the votes of its closest neighbours. While this was applied only to MIDI data, short segments of musical pieces could have been used with the SMBGT for feature extraction [19].

In this work, we do not cover the fundamentals of data mining algorithms, but we recommend the following references in case further reading is required. Goodfellow et al. [20] give an introduction to a broad range of topics in deep learning, covering mathematical and conceptual aspects of different artificial neural network models used in industry and academia. Rokach and Maimon [21] present an in-depth overview of the many techniques based on decision trees used for data mining including an analysis of the specifics of the Random Forest algorithm. Finally, Hastie et al. cover the fundamentals of computational techniques for statistical learning, inference and prediction [22].

2.1. Business Understanding

This step involves understanding business goals, limitations and expectations. The main objective is to cast the business problem as one or more data science problems. Framing it in terms of expected value can allow decomposing the problem into data mining tasks to which well-studied methods exist [9]. In the following paragraphs, we describe how digital transformation has made critical the use of data-mining techniques in the music industry.

Digital technologies permeate every market operation, the music industry is not an exception. On the contrary, music businesses have quickly adapted to the changes of the digital revolution since the advent of MP3 and portable music players. But nowadays the challenge is greater, given the amount of music being released daily has sky-rocketed. A rough estimate suggests that tracks released daily in streaming platforms doubled from 2018 to 2019, while the weekly average time spent listening to music reported by IFPI is circa 18 hrs [5]. Being able to recognise genres or musical categories is an important functionality for any music business, especially for streaming platforms seeking to engage their users by providing customized musical experiences.

In practice, we would like to know what music our users prefer (even if they do not know it yet) and provide recommendations on that basis. Glenn McDonald, Spotify's leading data scientist mentioned in an interview that the company has used a complex algorithm capable of analysing and categorizing up to 60 million songs on a molecular level, and the micro-classifications now amount to 1,387 sub-genres in total [8]. By understanding the inherent composition of songs within a certain genre and their sub-genres, similarities can be identified in terms of features and then leveraged for genre-based recommendation systems. In this way, music streaming services provide a customised dashboard of new releases and artists in real-time.

The quality of music recommenders and music discovery technologies becomes a powerful differentiator when the music catalogue is similar in size across streaming platforms. Around a third of Spotify's listening time is spent listening to Spotify-curated playlists, while slightly more

than half of that amount goes on playlists personalised to each listener based on their listening history. Nevertheless, while these add-ons might be beneficial, they have raised ethical concerns around the role of streaming services as tastemakers by biasing the visibility of certain artists [6].

Overall, based on the behaviour and profiles of customers, music features and information that is generated during playlist creation, companies can find patterns hidden within historical data and transform music recommendation and playlist curating into machine learning tasks such as classification, clustering, link prediction and association rule mining.

2.2. Data Understanding

Rarely there is an exact match between the data format required by data mining models and available data. Moreover, historical data might have been collected for purposes unrelated to our problem as framed in the business understanding step, or even for no explicit purpose at all. All of these situations have an impact on the reliability of data [9]. In the case of music genre recognition, data is obtained through licensing and partnerships with recording labels and independent artists. Any kind of automatic audio analysis relies on feature extraction to estimate meaningful patterns that are later fed to predictive and pattern recognition models. Audio feature extraction is the process of distilling huge amounts of raw audio into compact and high-level representations about the underlying musical content. Common choices for audio representations reflecting the way humans and many other organisms make sense of their auditory environment are wavelets, filterbanks and Fourier transforms. Nevertheless, more abstract features capture high-level aspects of music recording such as rhythm, harmony and timbral texture [7].

It is important to mention that audio is not the only source of music data. Other sources include metadata, lyrics, review, social tags, user profiles and playlists, MIDI files and music scores [7].

Audio tracks from the GTZAN collection were collected between 2000-2001 from a variety of sources including personal CDs, radio, microphone recordings, to represent a variety of recording conditions. All tracks are in WAV format, have a sampling rate of 22.5 kHz and a bit depth of 16. The 10 genres included in the database are Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, Rock.

Before the modelling stage, we carried out a series of exploratory analyses. Figure 2 shows the differences between the means of the spectral centroid feature across genres, while Figure 3 does the same for Chroma STFT. Since distributions overlap, we will need more than one feature to tell genres apart. Figure 4 shows the Spearman correlation between four of the features. Interestingly, some of the Mel frequency cepstral coefficients (MFCC) are inversely correlated with zero crossing. We decided not to show the Pearson correlation since the results were similar to Spearman. Finally, we carried out a t-test between genres in terms of the first MFCC. By using this feature, we can reliably tell apart Classical from the rest of the genres. ANOVA tests carried out for all the features of the data shows that the means of each group are significantly different from one another, Figure 6 presents the test carried out in terms of the ninth MFCC per genre. See the features list in the Data Preparation section for more information.

In addition to the aforementioned tests, we carried out a Shapiro-Wilk normality test per feature, where normality was the null hypothesis. All features but MFCC4, MFCC11, MFCC14, MFCC15, MFCC16, MFCC17 and MFCC18 had highly significant p-values. Finally, we computed a two-dimensional non-linear projection of the features using t-distributed stochastic neighbor embedding (TSNE) [14] (See Figure 1). The fact that genres are not perfectly segregated using a non-linear dimensionality reduction technique reflects the complexity of the

problem of genre classification. This gives us an idea of the predictive power of low-level features that ignore the contextual factors of music.

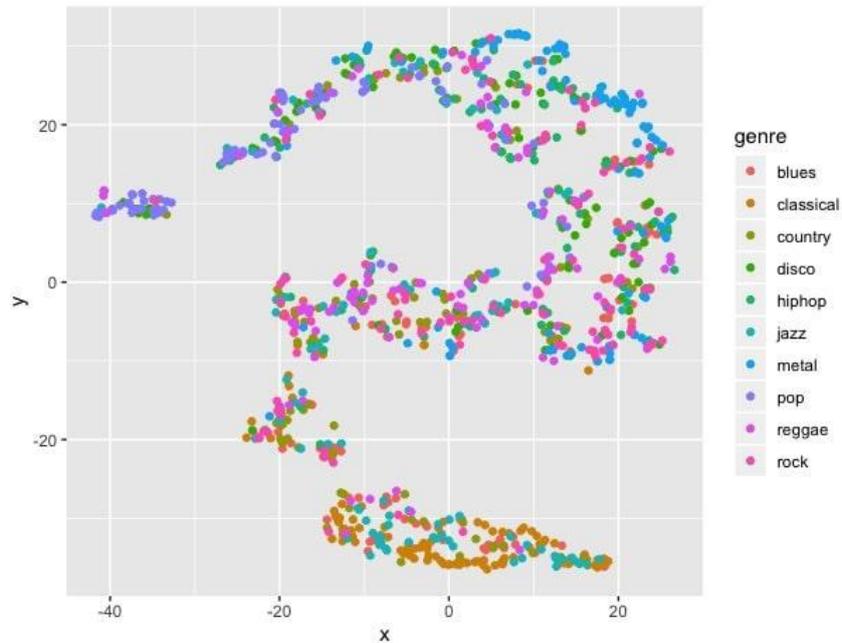


Figure 1. Dimensionality reduction using TSNE

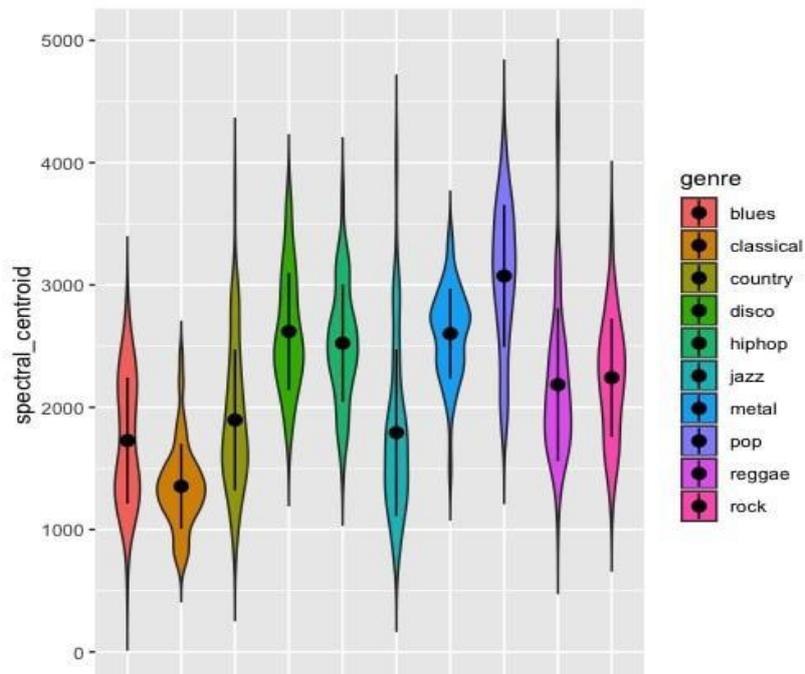


Figure 2. Spectral centroid distribution across genres

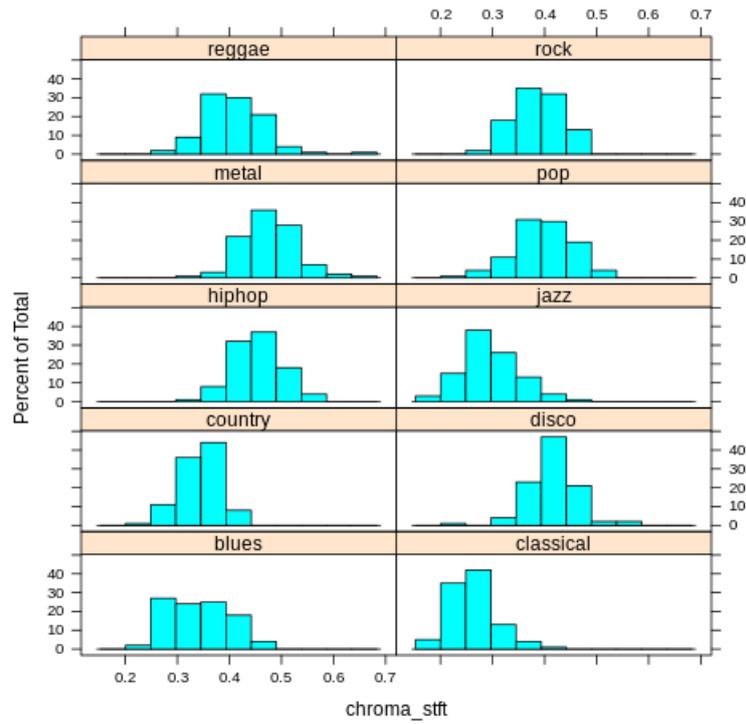


Figure 3. Chroma STFT comparison

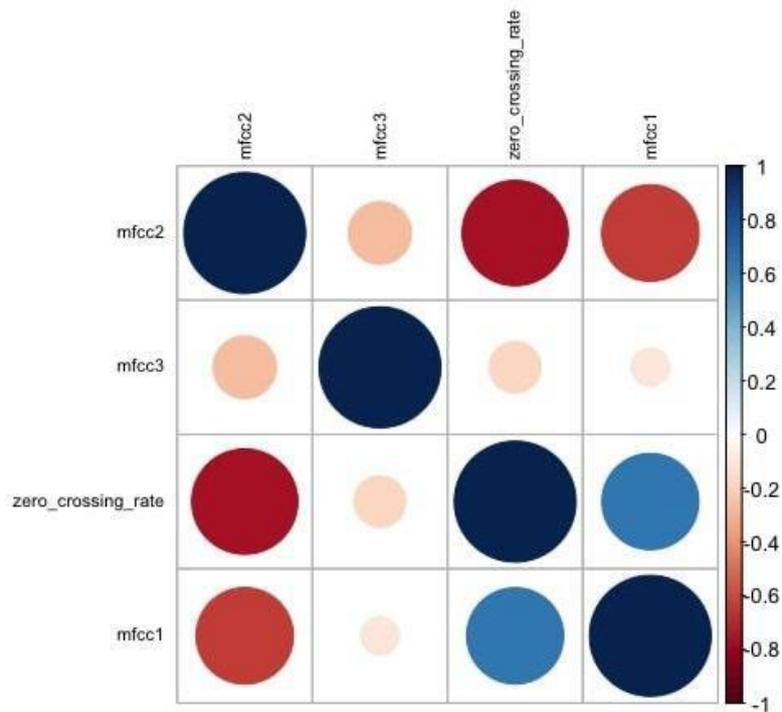


Figure 4. Spearman correlation of 4 features

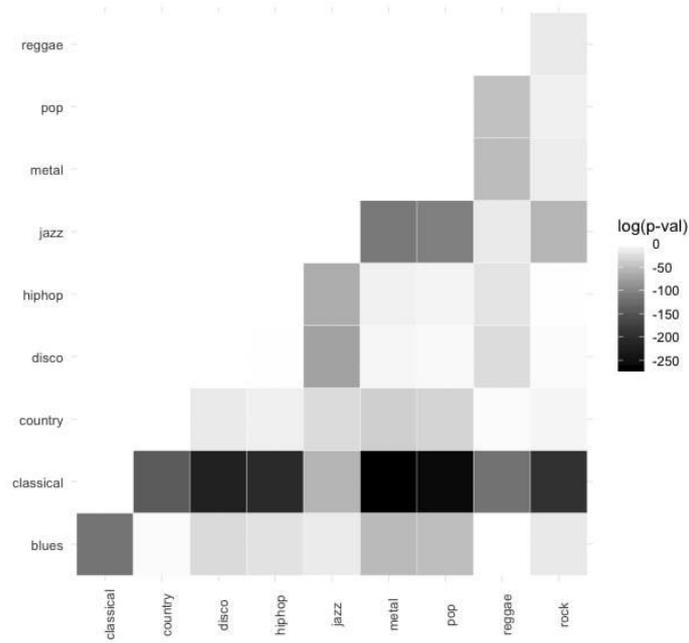


Figure 5. MFCC-1 t-test log(p-value) matrix

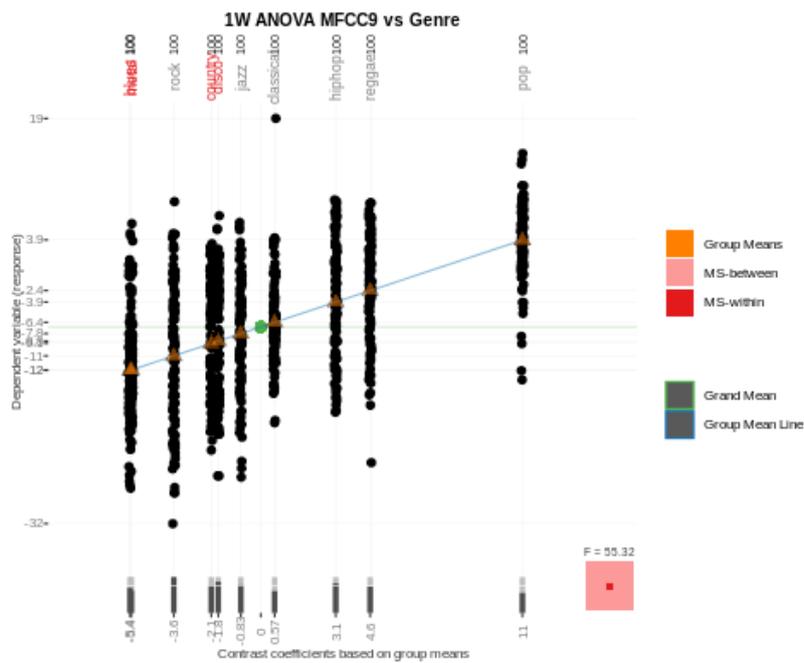


Figure 6. MFCC9 one-way ANOVA test

2.3. Data Preparation

We have to consider the nature of data before preparing it. For instance, some attributes may be categorical, whereas others text-based; All of these require different treatment. Overall, we have to make sure data is clean, free from leakages, and in the right format to carry out model fitting.

The 26 features used in this study were directly extracted from the audio tracks in GTZAN database using Librosa, a python package for music and audio analysis. A list with a short description is shown in the table below:

Table 1. Feature description

Feature	Description
Zero crossing rate	The rate of sign-changes along with a signal, i.e., the rate at which the signal changes from positive to negative or vice versa.
Spectral centroid	The “centre of mass” of the spectrum. Calculated as the weighted mean of the frequencies present in the sound
Spectral rolloff	Represents the frequency below which a specified percentage of the total spectral energy lies.
Root Mean Square Energy (RMSE)	Total magnitude of the signal.
Mel frequency cepstral coefficients (MFCCs)	A set of 20 features or coefficients describing the overall shape of the spectral envelope.
Chroma frequencies	A representation for music audio in which the entire spectrum is projected onto 12 bins accounting for the 12 distinct semitones (or chroma) of the musical octave.
2nd order Spectral Bandwidth	A weighted standard deviation from the spectral centroid

These 26 features were arranged in a table and appended to a .csv along with their corresponding filename and genre label.

Additionally, we used Wavenet as a feature extractor. These features are not human-readable but leverage the power of a convolutional neural network (CNN) trained directly on audio waveforms [13]. Since the shape of the feature map for each song was (125, 16) we also tested different aggregation methods including, flatten, column averages and row averages. These methods lead, respectively, to the following feature vector sizes: 2000, 125 and 16.

2.4. Modelling

We are concerned with the following data-mining tasks. These are related to recommendation operations in music streaming services. This paper covers model evaluation for the first two tasks.

- Classification: carry out class probability estimation for each song in our database to recommend more songs of users preferred genres.
- Clustering: Group songs by similarity to create sub-genres and fine-grained categories within and between genres.
- Link Prediction: attempt to predict connections between genres likeability. For instance, estimating how likely it is that someone who likes rock will like metal.
- Association rule mining: As part of the pattern extraction of songs, this task would help to understand how different types of genres and particular songs are grouped altogether by users and playlists.

For the first phase of the modelling step, we implemented two linear models: (I) a multinomial logistic regression to estimate genre (class) probability from the set of 20 Melfrequency cepstral coefficients (MFCCs), and (II) a generalized linear model to individually predict the respective values of Zero crossing rate, spectral centroid, spectral rolloff and RMSE from the nominal genre variable.

The multinomial logistic regression coefficients shown in the attached file are far from zero. This confirms our intuition that there was at least a linear relationship between predictors (MFCCs) and the probability of being of a certain genre. Differences across MFCCs weights reflect how the coefficients account for local variations in the spectral envelope that are correlated with the acoustic properties of genres.

With the linear models, we seek to test the hypothesis that the genres are meaningfully related to audio features, in particular those that describe musical and acoustic characteristics of the waveform such as the chroma frequencies, the centre of mass of the spectrum, etc. By generating a set of contrast variables using the genres and then fitting a set of linear models to predict each of the variables that do not fit under the umbrella of MFCCs, we corroborated this assumption. Moreover, the t-test showed significant differences across genres in terms of these covariates. To corroborate the generalised linear hypothesis of each model, a set of pairwise contrast variables was created using all the possible combinations of genres. These were tested using the Tukey method, again results show statistically significant differences across genres. Proving, further, the relation between features and genres.

Table 2. The architecture of the Fully Connected Network used for classifying the dataset.

Layer (type)	Output shape	Param #
Flatten	(None, 26)	0
Dense 1	(None, 32)	864
Dense 2	(None, 32)	1056
Dense 3	(None, 32)	1056
Dense 4	(None, 32)	1056
Dense 5	(None, 32)	1056
Dense 6	(None, 32)	1056
Dense 7	(None, 10)	330
Total trainable params	-	6,474

For the second phase, we explored the usage of more complex models for classifying the different music genres: A Fully Connected Network (FCN) described in Table2, and a family of Random Forest models. We compared their performance using thirty randomly drawn 70-30% training-testing splits. As a means of comparison, since the data is non-linearly separable, we fitted two logistic regression models using the same splits. Below we describe the implementation details of the models.

- 1) Fully-connected neural network (FCN): Implemented in Keras framework. The architecture consists of 6 layers with 32 units each. It was trained with RMSProp optimizer (default parameters) for 100 epochs and using a batch size of 32. See Figure 7. FCN-WF, FCN-WC and FCN-WR refer to the results of FCN using flattened, column-averaged and row-averaged Wavenet features, respectively.
- 2) Random Forest 1 (RF1): Implemented using sklearn ensemble classifiers. The model consists of 10,000 estimators trained over the dataset using Gini's impurity as splitting criteria.
- 3) Random Forest 2 (RF2): Preserves the same parameters as the previous model, only changing the splitting criterion from Gini's impurity to Information gain.
- 4) Random Forest 3 (RF3): Based on RF1 this model incorporated Minimal Cost-Complexity Pruning (CCP) to reduce overfitting. Only estimators with a CCP < 0.15 were selected to be part of the model.
- 5) Random Forests with Wavenet features: Following the same models as RF1 and RF3. Each pair of RF-WR, RF-WC and RF-WF show the results of using row-averaged, column-averaged and flattened Wavenet features, respectively.
- 6) Linear Regression model (LIN1): A regularised logistic regression model based on a quasi-Newton method (LBFGS) and Ridge Regression (L2) as implemented in sklearn.
- 7) Linear Regression model (LIN2): Similar to LIN1 but using an improved version of the Stochastic Average Gradient and Elastic net regularisation.

To test whether accuracy was significantly different across models, we carried out a Wilcoxon signed-rank paired test between the accuracy series. These were generated by evaluating the performance of trained models on 30 Test sets generated with the same random seeds (i.e. paired test). Figure 8 reports the p-values and Table 3 the performance results for each model.

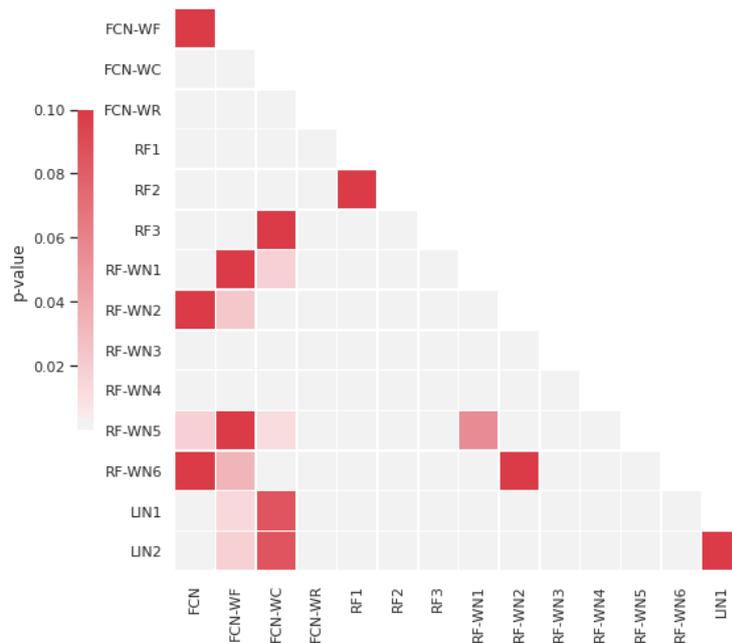


Figure 8. Wilcoxon signed-rank paired test for Test set

2.5. Evaluation

The purpose of the evaluation stage is to assess the data mining results and gain confidence that they are valid and reliable. These aspects include assessing models stability across time, ensuring that models satisfy the original business goals and spotting spurious correlations [9].

Even if we know from our musical experience there are systematic relationships across genres, the residual deviance of multinomial logistic regression is relatively high (2367.332), which means a linear model is a bad fit. Nevertheless, Non-parametric models such as Random Forest performed better than linear models and the fully-connected network (FCN). See Table 1.

By the other side, we believe some of the standard errors for multinomial logistic regression parameters are high, given that the sample size per genre is small and there is multicollinearity in our predictors, which means some MFCCs can be predicted from one another.

Table 3. Results: Mean accuracy for all models.

Model	Training %	Test %	Delta %
FCN	44.07	38.76	5.32
FCN-WF	45.46	40.66	4.81
FCN-WC	64.38	50.43	13.95
FCN-WR	72.82	56.34	16.48
RF1	99.93	64.48	35.45
RF2	99.93	64.19	35.74
RF3	61.18	49.38	11.80
RF-WR1	99.91	42.40	57.51
RF-WR2	48.74	37.49	11.25
RF-WC1	99.91	32.56	67.35
RF-WC2	47.94	24.84	23.10
RF-WF1	99.91	41.66	58.25
RF-WF2	80.15	37.68	42.47
LIN1	48.65	45.03	3.61
LIN2	48.81	44.71	4.10

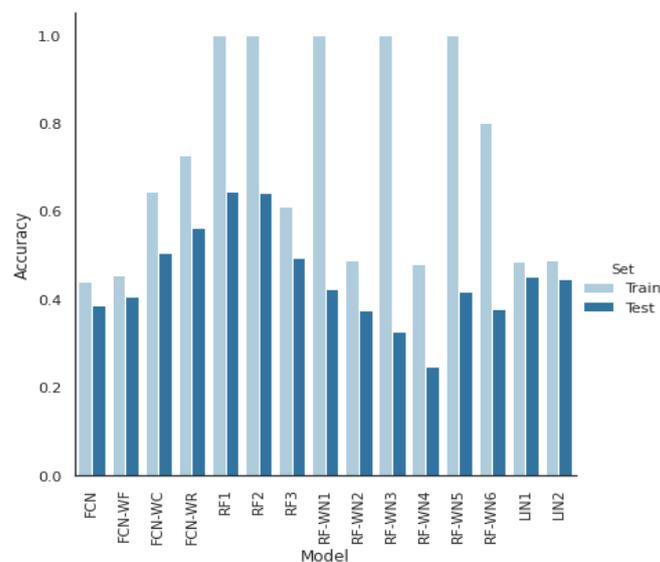


Figure 9. Mean accuracy comparison for Training and Test sets.

3. DISCUSSION

The main goal of this project is to identify characteristics within music genres to gain a better understanding of what patterns characterise them and which are shared amongst classes (i.e. genres). With this knowledge, we expect to be able to identify new songs that belong to a specific genre according to acoustic features and also to find sub-genres within them by using clustering techniques.

We noticed that some MFCCs are correlated among themselves but inversely correlated with other frequency related features such as zero crossings. The differences between genres in terms of MFCC1 were enough to tell apart Classical from the rest of the genres. During the data visualisation phase, we found that Pearson and Spearman's correlations gave similar results. For this reason, we decided to only include the latter.

Overall, the performance of the linear models, especially for class probability estimation, is low. Non-parametric models such as Random Forest performed better than linear models but overfit the training dataset. No significant differences were found between RF1 and RF2, which means for this task there is no noticeable advantage between Gini's impurity and information gain criteria. Despite the fully-connected network leveraging non-linear activation functions (ReLU), its accuracy was below the best Random Forest (RF1) and linear models. Interestingly, as opposed to Random Forests, the FCN accuracy improved using Wavenet features instead of handcrafted ones (i.e. MFCC, zero-crossings, etc.).

We also observed that the aggregation method (flatten, column average, row average) influenced the performance with Wavenet features for both Fully-connected neural networks and Random Forests. This might be explained by the fact using fewer features, as obtained with the column average method, reduces overfitting.

We confirmed that more expressive models than logistic regression, such as random forest, performed better at the classification task, but still are far from the required performance expected for music recommendation tasks (maximum average Test accuracy was 64%). Nevertheless, it was a good exercise to evaluate models with different assumptions: linear, non-parametric and hierarchical non-linear (i.e. artificial neural networks) with handcrafted and automatically extracted features (i.e. Wavenet). In this way, we compared how much improvement is coming from modelling non-linear interactions of covariates and learning structure from data without distributional assumptions.

4. CONCLUSION

Streaming services have disrupted the music industry, from how it is distributed to the way music is produced. Data mining techniques underlying this transformation have allowed companies to extract features with increasing precision and to use them for various purposes such as music recommendation, classification, trend prediction, sub-genre discovery and to design tailored user experiences for streaming apps.

We have demonstrated that there are significant differences in the feature values of each genre, seemingly enough to characterise each of them. The different tested linear models, leaving aside their poor performance, show that there is a strong relationship between the response (genre) and the proposed predictive variables. Furthermore, we found that non-parametric models, such as Random Forest with MFCC features, performed better than linear and Fully-connected neural

network models using Wavenet features. Our best model was a Random Forest (RF1) and achieved around 64% accuracy in the test set using handcrafted features (MFCC).

Despite not having met state-of-the-art results, the main contribution of this work is to compare a wide range of modern data mining techniques to study the dimensions of music, particularly its genre and the differences that exist amongst them. Moreover, to our best knowledge, we are the first to evaluate the effect of Wavenet features in the context of music genre recognition, particularly in the GTZAN database. The stark contrast between our models and state of the art approaches speaks about the relevance of finding good representations of musical data.

4.1. Future Work

Potential avenues for extending this work are: (1) exploring unsupervised clustering models, (2) multimodal models incorporating extra-musical and non-audio data, (3) transfer learning, data augmentation, and (4) extending GTZAN with new labelled instances (i.e. use a larger database). Clustering might find sub-structures within genres and help to determine useful patterns for music recommendation. Multimodal models [1] can leverage extra-musical information while transfer learning might be capable of finding better audio representations than MFCCs. Finally, we believe model training and testing results can be improved by applying other data mining and statistical techniques including but not restricted to LSTM neural networks, Dynamic Time Warping, K-means and Siamese Neural Networks.

REFERENCES

- [1] TadasBaltrusaitis, Chaitanya Ahuja, and Louis Philippe Morency. Multimodal Machine Learning: A Survey and Taxonomy, 2019.
- [2] Mehdi Banitalebi-Dehkordi and Amin Banitalebi-Dehkordi. Music genre classification using spectral analysis and sparse representation of the signals. *Journal of Signal Processing Systems*, 2014.
- [3] Grzegorz Gwardys and Daniel Grzywczak. Deep image features in music information retrieval. *International Journal of Electronics and Telecommunications*, 2014.
- [4] H. Honing, W.T. Fitch, B. Merker, I. Morley, W. Zuidema, L. Trainor, A. Patel, S.E. Trehub, J. Becker, M. Hoeschele, et al. *The Origins of Musicality*. The MIT Press. MIT Press, 2018.
- [5] IFPI. Ifpi releases music listening 2019, Sep 2019.
- [6] Mansoor Iqbal. Spotify usage and revenue statistics (2019), May 2019.
- [7] Tao Li, MitsunoriOgihara, and George Tzanetakis. *Music Data Mining*. CRC Press, Inc., USA, 1st edition, 2011.
- [8] Nick Patch. Meet the man classifying every genre of music on spotify - all 1,387 of them, Jan 2016.
- [9] Foster Provost and Tom Fawcett. *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*, 2013.
- [10] Colin Shearer. The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4), 2000.
- [11] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 2002.
- [12] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.
- [13] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, NalKalchbrenner, Andrew Senior, and KorayKavukcuoglu. Wavenet: A generative model for raw audio, 2016.
- [14] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.
- [15] Hansi Yang and Wei Qiang Zhang. Music genre classification using duplicated convolutional layers in neural networks. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019.

- [16] Hansi Yang and Wei Qiang Zhang. Music genre classification using duplicated convolutional layers in neural networks. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2019.
- [17] Alexander Schindler and Andreas Rauber. Capturing the temporal domain in echonest features for improved classification effectiveness. In Andreas Nürnberger, Sebastian Stober, Birger Larsen, and Marcin Detyniecki, editors, Adaptive Multi-media Retrieval: Semantics, Context, and Adaptation, pages 214–227, Cham, 2014.
- [18] Tristan Jehan and David DesRoches. Analyzer documentation (analyzer version 3.08). Website, 2011. Available online at http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf; visited on May 30th 2020.
- [19] AlexiosKotsifakos, Evangelos E. Kotsifakos, Panagiotis Papapetrou, and VassilisAthitsos. Genre classification of symbolic music with smbgt. In Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments, 2013.
- [20] Ian Goodfellow, YoshuaBengio, and Aaron Courville. 2016. Deep Learning. The MIT Press.
- [21] LiorRokach and Oded Maimon. Data mining with decision trees. Theory and applications, volume 69. 01 2008.
- [22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Elements of Statistical Learning. Springer, New York, NY, USA, 2008.

AUTHORS

Santiago Renteria is a computer scientist and audio engineer working at the intersection of artificial intelligence and biology. As part of his masters, he developed a “Shazam” for birdsong based on siamese neural networks, a few-shot machine learning technique capable of recognizing birds’ complex melodic sequences. Beyond machine learning one of his main interests is to develop and understand non-human forms of intelligence through artistic experimentation. In his practice, he plays with different media such as virtual reality, immersive audio and biosignal sensors. Currently, he is part of WATS, an interdisciplinary initiative generating projects at the interface of art, technology, and science. Furthermore, his work as a creative developer has been showcased at Laboratorio de Arte Alameda, Centro Cultural Universitario Tlatelolco, Carnaval de Bahidorá and Tecnológico de Monterrey.



Jesus Leopoldo Llano is a computer scientist whose work has mainly focused on the area of evolutionary computing and multi-objective optimisation. During his masters, he designed an evolutionary algorithm for the numerical treatment of equality constrained multi-objective optimization problems. His areas of professional interest focus mainly on the study and development of bio-inspired algorithms and computer systems as ways to solve high complexity problems. Currently, he is part of the machine learning research group of Tecnológico de Monterrey. His work has been showcased at the 50th Research and Innovation Congress of Tecnológico de Monterrey.



Francisco J. Cantu-Ortiz is Professor of Computer Science and Artificial Intelligence at Tecnológico de Monterrey. He is member of the Advisory Board for QS-World University Rankings and associate editor of various journals and conferences. His research interests include data science, AI analytics, science & technology management, and philosophy of science & religion. He has published more than 100 scientific documents and is a certified researcher by the National Council for Science and Technology, Mexico. He holds a PhD in Artificial Intelligence from the University of Edinburgh, UK, an MSc in Computer Science from North Dakota State University, USA, and a BSc in Computer Systems Engineering from Tecnológico de Monterrey (<http://semtech.mty.itesm.mx/fcantu/>)



(ITESM), México.

PREDICTION OF CANCER MICROARRAY AND DNA METHYLATION DATA USING NON-NEGATIVE MATRIX FACTORIZATION

Parth Patel¹, Kalpdrum Passi^{1§} and Chakresh Kumar Jain²

¹Department of Mathematics and Computer Science,
Laurentian University, Sudbury, Ontario, Canada

²Department of Biotechnology, Jaypee Institute of Information Technology,
Noida, India

ABSTRACT

Over the past few years, there has been a considerable spread of microarray technology in many biological patterns, particularly in those pertaining to cancer diseases like leukemia, prostate, colon cancer, etc. The primary bottleneck that one experiences in the proper understanding of such datasets lies in their dimensionality, and thus for an efficient and effective means of studying the same, a reduction in their dimension to a large extent is deemed necessary. This study is a bid to suggesting different algorithms and approaches for the reduction of dimensionality of such microarray datasets. This study exploits the matrix-like structure of such microarray data and uses a popular technique called Non-Negative Matrix Factorization (NMF) to reduce the dimensionality, primarily in the field of biological data. Classification accuracies are then compared for these algorithms. This technique gives an accuracy of 98%.

KEYWORDS

Microarray datasets, Feature Extraction, Feature Selection, Principal Component Analysis, Non-negative Matrix Factorization, Machine learning.

1. INTRODUCTION

There has been an exponential growth in the amount and quality of biologically inspired data which are sourced from numerous experiments done across the world. If properly interpreted and analyzed, these data can be the key to solving complex problems related to healthcare. One important class of biological data used for analysis very widely is DNA microarray data, which is a commonly used technology for genome-wide expression profiling [1]. The microarray data is stored in the form of a matrix with each row representing a gene and columns representing samples, thus each element shows the expression level of a gene in a sample [2]. Gene expression is pivotal in the context of explaining most biological processes. Thus, any change within it can alter the normal working of a body in many ways and they are key to mutations [3]. Thus, studying microarray data from DNA can be a potential method for the identification of many ailments within human beings, which are otherwise hard to detect. However, due to the large size of these datasets, the complete analysis of microarray data is very complex [4]. This requires some initial pre-processing steps for reducing the dimension of the datasets without losing information.

Modern technology has made it possible to gather genetic expression data easier and cheaper from microarrays. One potential application for this technology is in identifying the presence and stage of complex diseases within an expression. Such an application is discussed further in this study.

This study, reflects upon the Non-Negative Matrix Factorization (NMF) technique which is a promising tool in cases of fields with only positive values and assess its effectiveness in the context of biological and specifically DNA microarray and methylation data. The results obtained are also compared with the Principal Component Analysis (PCA) algorithm to get relative estimates.

Motivation for the proposed work was two-fold, first, to test the use of Non-negative Matrix Factorization (NMF) as a feature selection method on microarray and methylation datasets and optimize the performance of the classifiers on reduced datasets. Although several feature selection methods, e.g. Principal Component Analysis (PCA), and others have been used in literature, there is limited use and testing of matrix factorization techniques in the domain of microarray and methylation datasets.

2. RELATED WORK

Extensive presence of DNA microarray data has resulted in undertaking lots of studies related to the analysis of these data, and their relation to several diseases, particularly cancer. The ability of DNA microarray data over other such data lies in the fact that this data can be used to track the level of expressions of thousands of genes. These methods have been widely used as a basis for the classification of cancer.

Golub et al. proposed in [5] a mechanism for the identification of new cancer classes and the consecutive assignment of tumors to known classes. In the study, the approach of gene expression monitoring from DNA microarray was used for cancer classification and was applied to an acute leukemia dataset for validation purposes.

Ramaswamy et al. [6] presented a study of 218 different tumor samples, which spanned across 14 different tumor types. The data consisted of more than 16000 genes and their expression levels were used. A support vector machine (SVM)-based algorithm was used for the training of the model. For reducing the dimensions of the large dataset, a variational filter was used which in its truest essence, excluded the genes which had marginal variability across different tumor samples. On testing and validating the classification model, an overall accuracy of 78% was obtained, which even though was not considered enough for confident predictions, but was rather taken as an indication for the applicability of such studies for the classification of tumor types in cancers leading to better treatment strategies.

Wang et al. [7] provided a study for the selection of genes from DNA microarray data for the classification of cancer using machine learning algorithms. However, as suggested by Koschmieder et al. in [4], due to the large size of such data, choosing only relevant genes as features or variables for the present context remained a problem. Wang et al. [12] thus systematically investigated several feature selection algorithms for the reduction of dimensionality. Using a mixture of these feature selections and several machine learning algorithms like decision trees, Naive Bayes etc, and testing them on datasets concerning acute leukemia and diffuse B-cell lymphoma, results that showed high confidence were obtained.

Sorlie et al. (2001) [8] performed a study for the classification of breast carcinomas tumours using genetic expression patterns which were derived from cDNA microarray experiments. The

later part of the study also studied the clinical effects and implications of this by correlating the characteristics of the tumours to their respective outcomes. In the study, 85 different instances were studied which constituted of 78 different cancers. Hierarchical clustering was used for classification purposes. The obtained accuracy was about 75% for the entire dataset, and by using different sample sets for testing purposes.

Liu et al. [9] implemented an unsupervised classification method for the discovery of classes from microarray data. Their study was based on previous literature and the common application of microarray datasets which identified genes and classified them based on their expression levels, by assigning a weight to each individual gene. They modeled their unsupervised algorithm based on the Wilcoxon rank sum test (WT) for the identification and discovery of more than two classes from gene data.

Matrix Factorization (MF) is an unprecedented class of methods that provides a set of legal approaches to identify low-dimensional structures while retaining as much information as possible from the source data [22]. MF is also called matrix factorization, and the sequence problem is called discretization [22]. Mathematical and technical descriptions of the MF [19-20] methods, as well as microarray data [21] for their applications, are found in other reviews. Since the advent of sequencing technology, we have focused on the biological applications of MF technology and the interpretation of its results. [22] describes various MF methods used to analyze high-throughput data and compares the use of biological extracts from bulk and single-cell data. In our study we used specific samples and patterns for best visualization and accuracy.

Mohammad et al. [18] introduced a variational autoencoder (VAE) for unsupervised learning for dimensionality reduction in biomedical analysis of microarray datasets. VAE was compared with other dimensionality reduction methods such as PCA (principal components analysis), fastICA (independent components analysis), FA (feature analysis), NMF and LDA (latent Dirichlet allocation). Their results show an average accuracy of 85% with VAE and 70% with NMF for leukemia dataset whereas our experiments provide 98% accuracy with NMF. Similarly, for colon dataset, their results show an average accuracy of 68% with NMF and 88% with VAE whereas our experiments provide 90% accuracy with NMF.

2.1. Contributions

1. Two different types of high dimensional datasets were used, DNA Microarray Data (Leukemia dataset, prostate cancer dataset, colon cancer dataset) and DNA methylation Data (Oral cancer dataset, brain cancer dataset)
2. Two different feature extraction methods, NMF and PCA were applied for dimensionality reduction on datasets with small sample size and high dimensionality using different classification techniques (Random Forest, SVM, K-nearest neighbor, artificial neural networks).
3. All the parameters were tested for different implementations of NMF and classifiers and the best parameters were used for selecting the appropriate NMF algorithm.
4. Computing time Vs. number of iterations were compared for different algorithms using GPU and CPU for the best performance.
5. Results demonstrate that NMF provides the optimum features for a reduced dimensionality and gives best accuracy in predicting the cancer using various classifiers.

3. MATERIAL AND METHODS

3.1. Datasets

As a part of this study, five datasets relating to cancer were analyzed of which three are microarray datasets and two are methylation datasets. The aim of using methylation dataset is to ascertain the impact of DNA methylation on cancer development, particularly in the case of central nervous system tumors. The Prostate Cancer dataset contains a total of 102 samples and 2135 genes, out of which 52 expression patterns were tumor prostate specimens and 50 were normal specimens [6]. The second dataset used as a part of this study is a Leukemia microarray dataset [5]. This dataset contains a total of 47 samples, which are all from acute leukemia patients. All the samples are either acute lymphoblastic leukemia (ALL) type or of the acute myelogenous leukemia (AML) type. The third dataset used is a Colon Cancer microarray Dataset (Alon et al. 1999) [11]. This dataset contains a total of 62 samples, out of which 40 are tumor samples and the remaining 22 are from normal colon tissue samples. The dataset contained the expression samples for more than 2000 genes having the highest minimal intensity for a total of 62 tissues. The ordering of the genes was placed in decreasing order of their minimal intensities. The fourth and fifth dataset are the brain cancer and the oral cancer dataset with extensive DNA methylation observed. (Capper et al. 2018) [10]. the cancer methylation dataset is a combination of two types of information which are the acquired DNA methylation changes and the characteristics of cell origin. It is observed that such DNA methylation profiling is highly potent for the sub-classification of central nervous system tumors even in cases of poor-quality samples. The datasets are extensive with over 180 columns and 21000 rows denoting different patient cases. Classification is done using 1 for positive and 0 for negative instances.

3.2. Feature Selection

In the field of statistics and machine learning, feature reduction refers to the procedure for reducing the number of explanatory (independent) columns from the data under consideration. It is important in order to reduce the time and computational complexity of the model while also improving the robustness of the dataset by removal of correlated variables. In some cases, such as in the course of this study, it also leads to better visualization of the data.

Some of the popular feature reduction techniques include Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), Linear Discriminant Analysis (LDA), autoencoders, etc.

In this study, we analyze Non-negative Matrix Factorization (NMF) technique for feature selection and test the effectiveness in the context of biological and specifically DNA microarray and methylation data and compare our results with the PCA algorithm to get relative estimates.

3.2.1. Non-negative Matrix Factorization (NMF)

Kossenkov et al. (2010) [2] suggests the use of several matrix factorization methods for the same problem. The method presented in the paper is a Non-negative Matrix Factorization (NMF) technique which has been used for dimensionality reduction in numerous cases (Wang et al. 2013) [5].

For a random vector X with $(m \times n)$ dimensions, the aim of NMF is to try to express this vector X in terms of a basis matrix U ($m \times l$) dimension and a coefficient matrix V ($l \times n$) dimension, i.e.:

$$X \approx UV$$

The initial condition being $L \ll \min(m, n)$.

Here U and V are $m \times l$ and $l \times n$ dimensional matrices with positive values. U is known as the basis matrix while V is the coefficient matrix. The idea behind this algorithm is to obtain values of U and V such that the following function is at its local minima:

$$\min[D(X, UV) + R(U, V)]$$

where D : distance cost function, R : regularization function.

Results show that the obtained matrix U has a high dimension reduction from X .

On the application of the NMF algorithms, similar to other dimensionality reduction algorithms, a large number of variables are clustered together. In the case of NMF, expression data from the genes are reduced to form a small number of meta-genes (Brunet et al. 2004) [13]. For the matrix U , each column represents a metagene and the values give the contribution for the gene towards that metagene. The working of an NMF algorithm on a microarray dataset is shown in the figure below, with each pixel indicating the expression values (shown in the form of the intensity of colors).

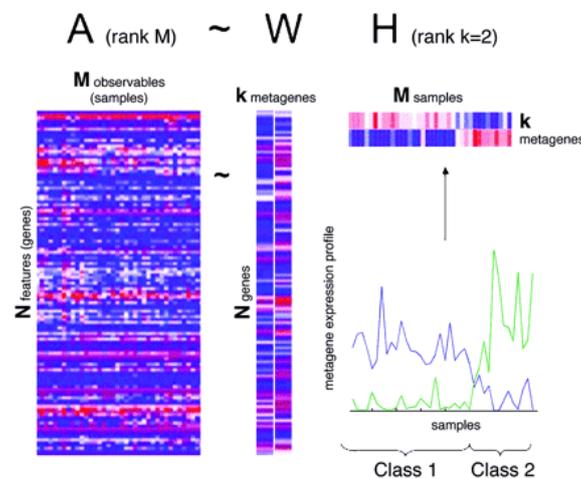


Figure 1. Image showing how an NMF algorithm is used to get a basis matrix of rank (2) [13]

After subsequent NMF based dimensionality reduction is done, the obtained reduced matrix is expected to contain the same information as the original matrix. In order to check the validity of the above assumption, classification algorithms are applied to the reduced matrix and subsequently, their accuracies were measured. Another popular dimensionality reduction technique which is the Principal Component Analysis (PCA) is used and then compared with the proposed NMF based models. It must be emphasized that conventional NMF based algorithms, even though very accurate are highly resource intensive when used for large datasets. This study, therefore, also makes use of certain GPU algorithms in order to effectively evaluate the same, making training time much more feasible.

There are different implementations of the NMF which are categorized based on the choice for the loss function (D) and the regularization function (R). The implementations used in this study are listed below.

- Non-smooth NMF (nsNMF) [14]
- Kullback Leibler Method (KL) [13]
- Frobenius [15]
- Offset [15]
- Multiplicative Update Algorithm (MU) [15]
- Alternating Least Square (ALS) [16]
- Alternating Constrained Least Square (ACLS) [17]
- Alternating Hoyer Constrained Least Square (AHCLS) [17]
- Gradient Descent Constrained Least Square (GDCLS) [15]

3.2.2. Principal Component Analysis (PCA)

PCA is a widely used technique in the field of data analysis, for orthogonal transformation-based feature reduction on high-dimensional data. Using PCA, a reduced number of orthogonal variables can be obtained which explain the maximum variation within the data. Using a reduced number of variables helps in significant reduction in computational cost and runtimes while still containing a high amount of information as within the original data.

It is a widely used tool throughout the field of analytics for feature reduction before predictive modelling. The steps involved in the computation of a PCA algorithm is shown below:

Let 'X' be the initial matrix of dimension (m x n), where m = number of rows, and n is the number of columns.

The first step is to linearly transform the matrix X into a matrix B such that,

$$B = Z * X$$

where, Z is a matrix of order (m x m).

The second step is to normalize the data. In order to normalize, the mean for the data is computed and normalization is done by subtracting off the mean for finding out the principal components. The equations are shown below:

$$M(m) = 1/N \sum_{n=1}^N X[m, n], X' = X - M$$

The next step involves computing the covariance matrix of X, which is computed as below

$$C_X = X * \frac{X^T}{(n-1)}$$

In the covariance matrix, all diagonal elements represent the variance while all non-diagonal elements represent co-variances.

The covariance equation for B is shown below

$$C_B = B * \frac{B^T}{(n-1)} = \frac{(ZA)(ZA)^T}{(n-1)} = \frac{(ZA)(A^T * Z^T)}{(n-1)} = \frac{ZY * Z^T}{(n-1)}$$

where $Y = A * A^T$, and of dimension (m x m),

Now, Y can be further expressed in the form, $Y = EDE$, where E is an orthogonal matrix whose columns represent the eigenvalues of Y, while D is a diagonal matrix with the eigenvalues as its entries.

If $Z = E^T$, the value of covariance for B becomes,

$$C_B = \frac{ZY*Z^T}{(n-1)} = \frac{E^T(ED*E^T)E}{(n-1)} = \frac{D}{(n-1)}$$

The eigenvalues in this case are arranged in descending order, thus the most important component comes first and so on.

Thus, from the transformed matrix 'B', only a subset of features can be taken which preserve a larger share of the variance within the data.

3.2.3. Parameter Selection

Table 1 summarizes the different parameters which have been used for different algorithms.

Table 1: Parameters

Algorithm	Parameters
Mu	Rank = 5
GDCLS	Rank = 5, $\lambda = 0.1$
ALS	Rank = 5
ACLS	Rank = 5, $\lambda_H = 0.1$, $\lambda_W = 0.1$
AHCLS	Rank = 5, $\lambda_H = 0.1$, $\lambda_W = 0.1$, $\alpha_H = 0.5$, $\alpha_W = 0.5$
PCA	Number of features = varying between 1 to 100. row.W = 1, col.W = 1
Random Forest	nTrees = 500, cutoff = $\frac{1}{2}$, nodesize = 1
SVM	coef = 0, cost = 1, nu = 0.5, tolerance = 0.001

3.2.4. Runtime Computation using CPU and GPU

There exist many different algorithms for implementation of NMF using R. These algorithms can be broadly classified as those which are implemented using the CPU architecture of the system and those which are implemented using GPU architectures. The CPU codes are implemented using the standard 'NMF' package within R while the GPU codes are implemented using a modification of the 'NMF' package known as the 'NMFGPU4R' package which uses multicore options from GPUs to massively parallelize the implementation of the algorithms.

The different algorithms are first run on the three datasets and their run times are noted respectively. The runtime is defined as the amount of time (in seconds) taken by the computer to compute that respective algorithm. A higher runtime generally means high complexity and should be avoided, as such algorithms generally don't scale very well. During each of the implementation, we chose three clusters within the output dataset in order to provide uniformity.

For the algorithms executed on CPUs, the obtained run times are given in Table 2.

Table 2. CPU times (seconds) of different NMF algorithms on the cancer datasets

Method	Prostrate	Colon	Leukemia
NsNMF	90.43	169.19	157.1
KL	9.85	8.29	6.08
Frobenius	3.86	4.16	3.09
Offset	103.63	164.62	160.50

As can be seen clearly for the three datasets, the *Offset* method takes the highest runtime of over 100s for prostate cancer data and over 160s for colon cancer and leukemia data, closely preceded by the nsNMF method which is just under 160s. The Frobenius and the KL methods take significantly lesser times (under 10s) than the other two methods. These algorithms could not be computed on the methylation datasets using CPUs as these are very high dimensional (over 10000 rows) datasets.

For the algorithms executed on GPUs, the obtained run times are given in Table 3.

Table 3. GPU times (seconds) of different NMF algorithms on the cancer datasets

Method	Microarray dataset	Brain Cancer	Oral Cancer
Mu	0.66	5.01	4.18
ALS	0.77	1.823	1.20
GDCLS	1.44	0.34	0.309
NSNMF	0.96	4.334	4.114
ACLS	0.58	1.287	1.19
AHCLS	0.51	1.11	0.67

As can be seen from Table 3, the AHCLS and the ACLS are the quickest to converge with runtimes of 0.51s and 0.58s respectively. However, the brain and oral cancer datasets had different outcomes in this regard. The GDCLS method was the quickest to converge for brain and oral cancer datasets with a runtime of 0.34s and 0.309s respectively.

3.3. Classification Algorithms

The classification algorithms used in this study include random forest, support vector machine, neural networks and k-nearest neighbors. These are briefly explained.

3.3.1. Random Forest (RF)

The random forest algorithm is a supervised learning algorithm which is widely used as an ensemble model for classification and regression tasks. Classification involves training a model with a set of independent variables in order to output a dependent variable into certain pre-defined factors. The random forest algorithm is an ensemble of decision trees. The output is a mean of the output for the different decision trees. The training algorithm from random forest is based on bootstrap aggregating to tree classifiers. Generally, for a given training sample (x_1, x_2, \dots, x_n) along with their respective response variables (y_1, y_2, \dots, y_n) , for $b = 1, \dots, B$, random samples are selected with replacement from the n training examples and the classifiers are then trained on them using trees f_b .

The equation for prediction for out-of-sample data in case of a random classifier is shown below:

$$p = \frac{1}{B} \sum_{b=1}^B \mathcal{F}(x)$$

3.3.2. Support Vector Machine (SVM)

The Support Vector Machine algorithm is a very commonly used algorithm for classification and regression. The models is a non-probabilistic kind where each individual data point is represented in an n-dimensional space where n refers to the number of independent features. Classification is done by fitting an (n-1) dimensional plane within the space and on the basis of the position of a particular data point with respect to the plane. The confidence of prediction is determined by the distance of the position of the point with respect to the dividing plane.

Let $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ be a set of n points where x represents a p-dimensional vector and y is an indicator variable for the class which takes the value 0 or 1. The goal of SVM is to find a dividing hyperplane that divides the entire n-points based on their indicator variable value. Such a hyperplane can be represented as

$$\underline{w} \cdot \underline{x} - b = 0$$

where, w is the normal vector and, b is a parameter which determines the offset for the hyperplane.

3.3.3. Neural Network (NN)

Neural Networks comprise of a collection of different algorithms and paradigms which resemble the networks within the human brain in the context that they take input and process it through different layers and nodes. Inputs are passed on as vectors and the neural networks perform operations which help us cluster or regress this data.

The primary constituent of a neural network is a set of different layers Each layer is further made up of nodes. This entire structure is modelled based on the neural structure of a human brain. With each node is associated a set of weights and coefficients which largely determine the amplification of the amount of signal of data which passes through it. The final layer contains a single node which sums up the values from all the nodes in its preceding layer and outputs a single value indicating the label (classification) or the value of the output (regression).

3.3.4. K-nearest Neighbor (KNN)

The K-nearest Neighbor (KNN) model is a non-parametric model which is commonly used in the field of machine learning for the purpose of classification or regression tasks.

In this method, the input is a vector of some 'k' closest training examples which exist in the hyperspace and based on the value of these k-nearest neighbors, the classification for the object is done based on a majority voting about the classes of its neighbors.

4. RESULTS

In order to evaluate the performance of classification algorithms on the NMF and PCA reduced data four classifiers were trained, random forest (RF), support vector machine (SVM), neural network (ANN) and k-nearest neighbor (KNN). The classifiers were tested for accuracy and area under the ROC curve (AUC) performance metrics. 10-fold cross validation was used to avoid overfitting and different number of features were selected using NMF and PCA methods. Figure 2 shows the accuracy of the four classifiers with NMF and PCA methods on the leukemia dataset.

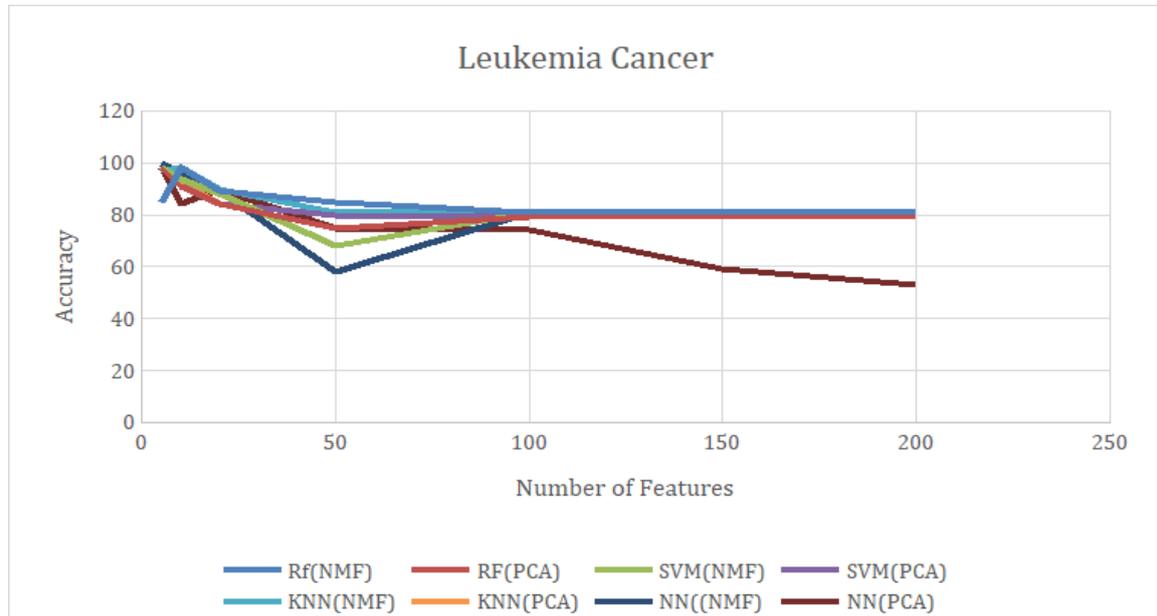


Figure 2. Accuracy of classifiers with different number of features for leukemia dataset

It is evident from Figure 2 that the accuracies obtained for lower number of features (10) are the most accurate with about 98% accuracy obtained for most classifiers in case of NMF and a very high AUC (~ 0.97). For PCA reduced matrices, the highest accuracy is also in the range of 98% but for lesser number of features (5).

Figure 3 shows the accuracy of the four classifiers with NMF and PCA methods on the prostate dataset.

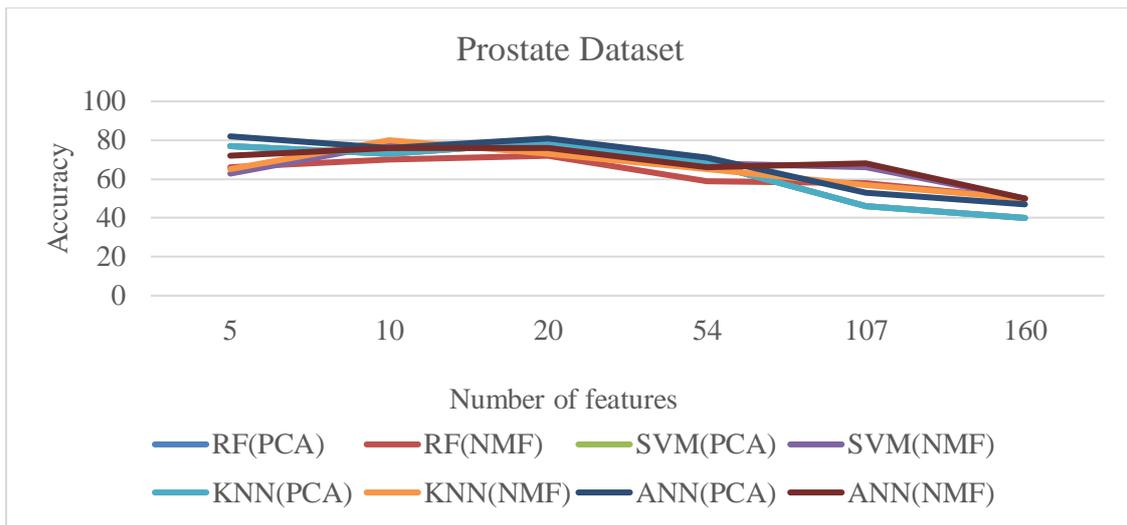


Figure 3. Accuracy of classifiers with different number of features for prostate dataset

It is evident from Figure 3 that the accuracy generally decreases as the number (percentage) of features increase. For lesser number of features (10), the accuracy is in the range of 70-80%. However, as the features increase, the accuracy falls to about 50%. The AUC values also show a similar trend.

Figure 4 shows the accuracy of the four classifiers with NMF and PCA methods on the colon dataset.

In the case of colon dataset, the SVM classifier gives an accuracy in the range of about 87% for lower number of features (10). As in the case of other datasets, there is a general decrease in the accuracy as the number of features increases.

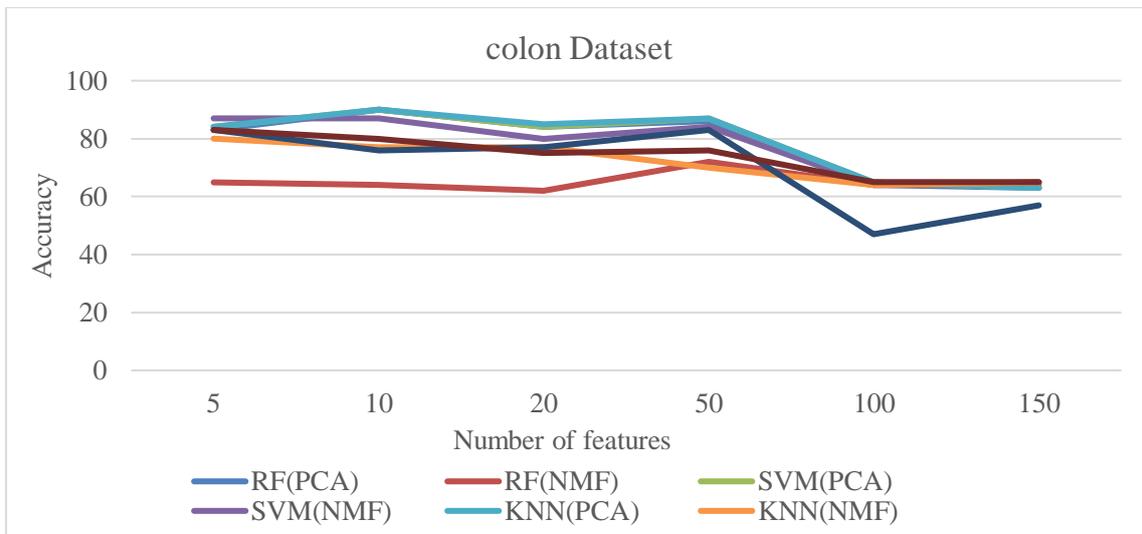


Figure 4. Accuracy of classifiers with different number of features for colon dataset

Figure 5 shows the accuracy of the four classifiers with NMF and PCA methods on the oral cancer dataset.

The Oral Cancer dataset has a general low accuracy across all classifiers (~65%) for all the different combinations of number of features. However, the AUC values are high (~0.95).

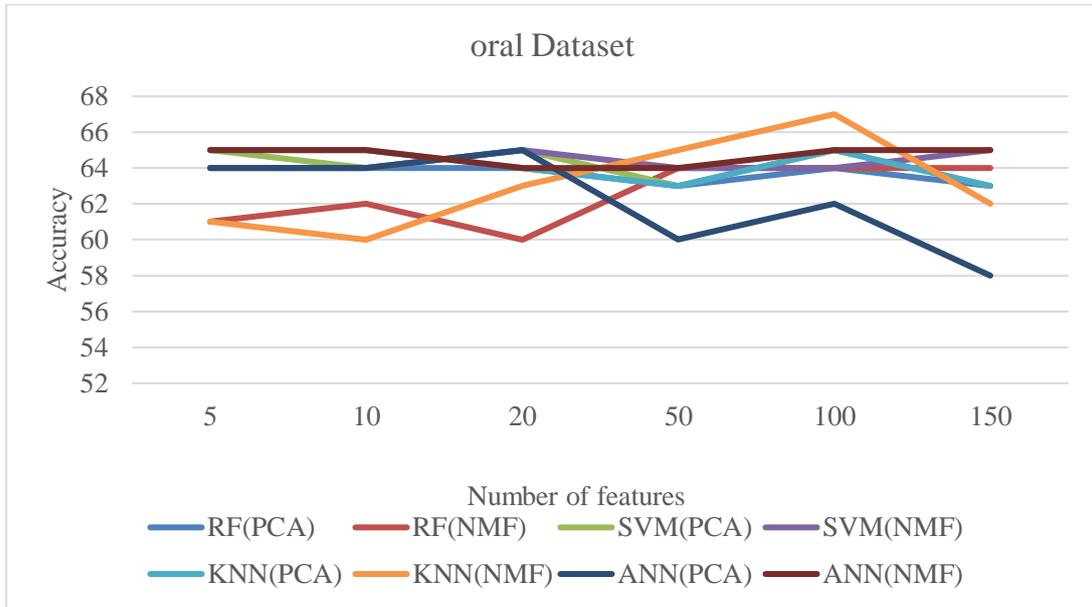


Figure 5. Accuracy of classifiers with different number of features for oral cancer dataset

Figure 6 shows the accuracy of the four classifiers with NMF and PCA methods on the brain cancer dataset.

Both the NMF and the PCA algorithms perform similarly in this case with the highest accuracy being 92% and 95% respectively. The ANN algorithm shows a consistent performance of over 90% in this case.

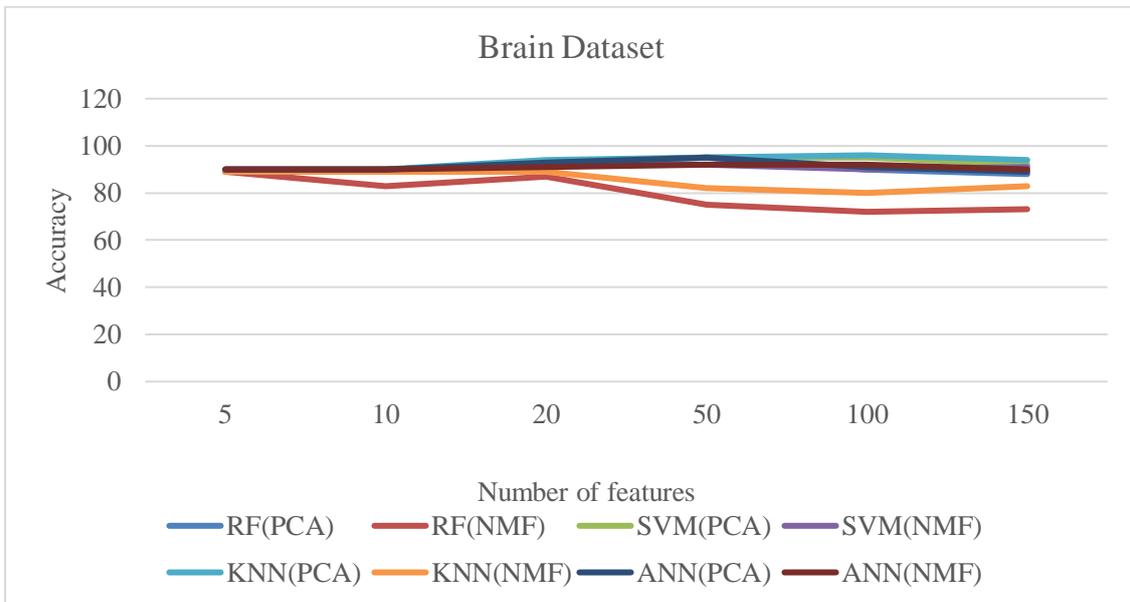


Figure 6. Accuracy of classifiers with different number of features for brain cancer dataset

For the second phase of our analysis on the classification algorithms, different training-to-testing ratios were taken to find the best accuracy and AUC values.

For the leukemia dataset, it is observed that for lower number of features, the ANN method is the best in terms of accuracy (~99%) as opposed to over 80% shown by other classifiers. However, as the number of features increase the accuracy for all the cases decreases. In the case for the Prostate Cancer dataset, accuracies in the range of 70-90% are obtained throughout the different combinations. The highest accuracy is obtained for the training-to-testing ratio 50:50 while using ANN for a small number of features (5). The accuracy in this case is close to 100%. For the colon cancer dataset, it is seen that the raw data has the best accuracy for the RF and SVM models while the highest accuracy is obtained for the ANN model (95%). The brain cancer dataset is the first of methylation datasets. It is seen that the performance of the two algorithms is consistent over the range of split percentages i.e. (~90%), for the Random Forest classifier with reduced features (5). The ANN also performs with a similar accuracy along with the KNN model (accuracy ~89%). A lower accuracy is obtained for the oral cancer dataset across all combinations. The overall range for the accuracy is between 60-80% barring a few exceptions. The ANN is the most consistent classifier here with classification accuracies above 80% in all other cases except when the number of reduced features is the highest (5).

Table 4 shows the summary results of the best accuracies for each classifier and dataset for NMF and PCA algorithms. After performing all the analysis, the following points become pertinent:

1. Random Forest and KNN classifiers have an accuracy of 98% after using just 5 and 10 of the number features in the leukemia dataset for the NMF and PCA reduced datasets respectively.
2. SVM algorithm also gives 98% accuracy for the leukemia dataset with only 5 features.
3. ANN algorithm gives 100% accuracy with 5 features used with NMF. However, the accuracy for the other datasets are lower with the Brain Cancer dataset giving over 90% accuracy.
4. For the colon cancer dataset, PCA generally performs better than the NMF algorithm as RF, SVM and KNN have about 90% accuracy with PCA, however it is about 80% with NMF.
5. The PCA+RF and PCA+ANN combination in case of brain cancer data gives 95% accuracy with 50 features.
6. The oral cancer dataset has the lowest accuracies, with the highest being 69% using the PCA+SVM classifier on the Raw Data.

Table 4. Summary results of best accuracy and number of features

Classifier	Dataset	NMF		PCA	
		Acc.	# of features	Acc.	# of features
RF	Leukemia	98	10	98	5
	Prostate Cancer	72	20	77	5
	Colon Cancer	74	10	90	20
	Brain Cancer	89	5	95	50
	Oral Cancer	65	180	65	180
SVM	Leukemia	98	5	98	5
	Prostate Cancer	78	150	79	20
	Colon Cancer	87	5	90	10
	Brain Cancer	92	20	95	50
	Oral Cancer	66	180	69	180
KNN	Leukemia	98	10	98	5
	Prostate Cancer	80	10	79	20
	Colon Cancer	80	5	90	10
	Brain Cancer	89	10	96	100
	Oral Cancer	67	100	65	10
ANN	Leukemia	100	5	98	5
	Prostate Cancer	76	20	81	20
	Colon Cancer	83	5	83	5
	Brain Cancer	92	100	95	50
	Oral Cancer	65	150	65	180

5. CONCLUSIONS AND FUTURE WORK

In this study, NMF has been used on several microarray datasets in order to obtain dimensionality reduction in the feature sets. After application of NMF algorithms, the number of variables from the gene expression data using microarray profiles went down from about 1000 to a handful. On reduction, the obtained datasets were easier to read both visually and through heat maps and plots. An optimum number of reduced features were obtained and for this, individual heap maps and coefficient-maps were plotted. Unlike previous methods, where reducing the number of variables required extensive study on the nature of the datasets, using this approach, the same could be done through numerical computations on the given datasets.

This study is of immense biological significance. Its significance lies in detecting and identifying genes for differentiating cancer diseases and non-cancer diseases so that a proper tailor-made treatment can be initiated. It also recognizes specific biomarkers of cancer which can be further analyzed. It is essential to identify genes which play a role in the development of a cancer as the gene expressions of patients suffering from cancer are different from those of healthy patients. It also makes analyzing data easier for using machine learning techniques.

Throughout the course of the study there are two major drawbacks which are elucidated below. Due to limited computational resources, the entire set of NMF algorithms couldn't be run on every dataset, particularly the methylation ones which have higher dimensions, leading to certain imperfections in their analysis. This also resulted in our inability to perform semi-NMF calculations as well as residual analysis plots for these datasets. The lack of computational power was again the major reason for keeping the subset feature numbers restricted to 50. As any number of features higher than that would exponentially increase the training time for the classifiers.

As discussed above, NMF has potential use for feature reduction of high dimensional data, particularly in the context of microarray data. A significant work can be done in the direction of classifying different subgroups of a particular disease by first using NMF to reduce high dimensional microarray data. This not only simplifies the process but also provides a faster and more robust model of diagnosis.

Use of NMF for methylation datasets is limited in the context that a lower accuracy is observed for these cases. Further studies should revolve around improving the scenario of the same.

REFERENCES

- [1] A. Singh, "Microarray Analysis of the Genome-Wide Response to Iron Deficiency and Iron Reconstitution in the Cyanobacterium *SynchroSystems* sp. PCC 6803", *PLANT PHYSIOLOGY*, vol. 132, no. 4, pp. 1825-1839, 2003. Available: 10.1104/pp.103.024018.
- [2] A. Kossenkov and M. Ochs, "Matrix factorization methods applied in microarray data analysis", *International Journal of Data Mining and Bioinformatics*, vol. 4, no. 1, p. 72, 2010. Available: 10.1504/ijdm.2010.030968.
- [3] D. Nott et al., "Hierarchical Bayes variable selection and microarray experiments", *Journal of Multivariate Analysis*, vol. 98, no. 4, pp. 852-872, 2007. Available: 10.1016/j.jmva.2006.10.001.
- [4] A. Koschmieder, K. Zimmermann, S. Trissl, T. Stoltmann and U. Leser, "Tools for managing and analyzing microarray data", *Briefings in Bioinformatics*, vol. 13, no. 1, pp. 46-60, 2011. Available: 10.1093/bib/bbr010.
- [5] T. Golub, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring", *Science*, vol. 286, no. 5439, pp. 531-537, 1999. Available: 10.1126/science.286.5439.531.
- [6] S. Ramaswamy et al., "Multiclass cancer diagnosis using tumor gene expression signatures", *Proceedings of the National Academy of Sciences*, vol. 98, no. 26, pp. 15149-15154, 2001. Available: 10.1073/pnas.211566398.
- [7] Z. Wang et al., "Microarray Analysis of Gene Expression Involved in Another Development in rice (*Oryza sativa* L.)", *Plant Molecular Biology*, vol. 58, no. 5, pp. 721-737, 2005. Available: 10.1007/s11103-005-8267-4.
- [8] T. Sorlie et al., "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications", *Proceedings of the National Academy of Sciences*, vol. 98, no. 19, pp. 10869-10874, 2001. Available: 10.1073/pnas.191367098.
- [9] Y. Liu-Stratton, S. Roy and C. Sen, "DNA microarray technology in nutraceutical and food safety", *Toxicology Letters*, vol. 150, no. 1, pp. 29-42, 2004. Available: 10.1016/j.toxlet.2003.08.009.
- [10] D. Capper et al., "DNA methylation-based classification of central nervous system tumours", *Nature*, vol. 555, no. 7697, pp. 469-474, 2018. Available: 10.1038/nature26000.

- [11] U. Alon et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745-6750, 1999. Available: 10.1073/pnas.96.12.6745.
- [12] J. Wang, X. Wang and X. Gao, "Non-negative matrix factorization by maximizing correntropy for cancer clustering", *BMC Bioinformatics*, vol. 14, no. 1, p. 107, 2013. Available: 10.1186/1471-2105-14-107.
- [13] J. Brunet, P. Tamayo, T. Golub and J. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization", *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164-4169, 2004. Available: 10.1073/pnas.0308531101.
- [14] A. Pascual-Montano, J. Carazo, K. Kochi, D. Lehmann and R. Pascual-Marqui, "Non smooth nonnegative matrix factorization (nsNMF)", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403-415, 2006. Available: 10.1109/tpami.2006.60
- [15] C. Févotte and J. Idier, "Algorithms for Nonnegative Matrix Factorization with the β -Divergence", *Neural Computation*, vol. 23, no. 9, pp. 2421-2456, 2011. Available: 10.1162/neco_a_00168
- [16] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values", *Environ metrics*, vol. 5, no. 2, pp. 111-126, 1994. Available: 10.1002/env.3170050203
- [17] A. N. Langville, C. D. Meyer, R. Albright, J. Cox, and D. Duling, "Algorithms, initializations, and convergence for the nonnegative matrix factorization", *CoRR*, vol. abs/1407.7299, 2014.
- [18] Mohammad Sultan Mahmud ,Xianghua Fu,Joshua Zhexue Huang and Md.Abdul Masud "Biomedical Data Classification Using Variational Autoencoder ", @ Springer Nature Singapore pte Ltd.2019 R.Islam et al(Eds): Aus DM 2018 CCIS 996, pp.30-42,2019.
- [19] Ochs MF and Fertig EJ (2012) Matrix factorization for transcriptional regulatory network inference. *IEEE Symp. Comput. Intell. Bioinforma. Comput. Biol. Proc 2012*, 387–396 [PMC free article] [PubMed] [Google Scholar]
- [20] Li Y et al. (2016) A review on machine learning principles for multi-view biological data integration. *Brief. Bioinform* 19, 325–340 [PubMed] [Google Scholar]
- [21] Devarajan K (2008) Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS Comput. Biol* 4, e1000029 [PMC free article] [PubMed] [Google Scholar]
- [22] Stein-O'Brien, Genevieve L et al. "Enter the Matrix: Factorization Uncovers Knowledge from Omics." *Trends in genetics: TIG* vol. 34,10 (2018): 790-805. doi: 10.1016/j.tig.2018.07.003

AUTHORS

Parth Patel graduated from Laurentian University, Ontario, Canada with Master's degree in Computational Sciences. He pursued his bachelor's degree in Computer Engineering from the Gujarat Technological University, Gujarat, India. His main areas of interest are Machine Learning Techniques, Data Analytics and Data Optimization Techniques. He completed his M.Sc. thesis in Prediction of Cancer Microarray and DNA Methylation Data Using Non-Negative Matrix Factorization.



Kalpdrum Passi received his Ph.D. in Parallel Numerical Algorithms from Indian Institute of Technology, Delhi, India in 1993. He is an Associate Professor, Department of Mathematics & Computer Science, at Laurentian University, Ontario, Canada. He has published many papers on Parallel Numerical Algorithms in international journals and conferences. He has collaborative work with faculty in Canada and US and the work was tested on the CRAY XMP's and CRAY YMP's. He transitioned his research to web technology, and more recently has been involved in machine learning and data mining applications in bioinformatics, social media and other data science areas. He obtained funding from NSERC and Laurentian University for his research. He is a member of the ACM and IEEE Computer Society.



Chakresh Kumar Jain received his Ph.D. in the field of bioinformatics from Jiwaji University, Gwalior, India, focusing on computational designing of non-coding RNAs using machine learning methods. He is an Assistant Professor, Department of Biotechnology, Jaypee Institute of Information Technology, Noida, India. He is CSIR-UGC-NET [LS] qualified and member of International Association of Engineers (IAENG) and life member of IETE, New Delhi, India. His research interests include development of computational algorithms for quantification of biological features to decipher the complex biological phenomenon and diseases such as cancer and neurodegenerative diseases apart from drug target identification and mutational analysis for revealing the antibiotic resistance across the microbes through computer based docking, molecular modelling and dynamics, non-coding RNAs identification, machine learning, data analytics, and systemsbiology based approaches.



A FRAMEWORK FOR CAPTURING AND ANALYZING UNSTRUCTURED AND SEMI-STRUCTURED DATA FOR A KNOWLEDGE MANAGEMENT SYSTEM

Gerald Onwujekwe¹ Kweku-Muata Osei-Bryson¹
and Nnatubemugo Ngwum²

¹Department of Information Systems, Virginia Commonwealth University,
Richmond, VA, USA

²Department of Computer and Information Sciences,
Towson University, Maryland, USA

ABSTRACT

Mainstream knowledge management researchers generally agree that knowledge extracted from unstructured data and semi-structured data has become imperative for organizational strategic decision making. In this research, we develop a framework that captures and analyses unstructured data using machine learning techniques and integrates knowledge and insight gained from the data into traditional knowledge management systems. Unlike most frameworks published in the literature that focuses on a specific type of unstructured data, our frameworks cut across the varieties of unstructured data ranging from textual data from social network sites, online forums, discussion boards, reviews to audio data, image data and video data. We highlight some preprocessing and processing techniques for these data and also highlight some standard output. We evaluate the framework by developing a textual data application programming interface (API) using python and beautiful soup and we perform sentiment analysis on the students' review data collected through the API.

KEYWORDS

Unstructured data, knowledge management system, framework, sentiment analysis.

1. INTRODUCTION

Researchers have noted the organizational value of capturing unstructured and semi-structured knowledge. Ramesh and Tiwana [1] said that the business value of codification and capturing semi-structured knowledge was found in a recent study of 120 projects across a cross-section of firms. They further highlighted that the integration of the knowledge management system they developed with a groupware system for formal and informal interactions, is essential for a comprehensive representation of process knowledge. Knowledge management (KM) is a discipline that promotes an integrated approach to identifying, capturing, evaluating, retrieving, and sharing all the enterprise's information assets including databases, documents, and procedures, among others [2]. [3] has studied knowledge management, e-learning systems, and organizational learning as the three variables that impact organizational intelligence, with no attention paid to external sources of knowledge. Organizational intelligence which seeks to measure the ability of an organization to generate knowledge relevant to what the organization

does, should not be defined based on the knowledge that could be harnessed internally within the organization only. [4] observed that some companies, like Andersen Consulting and Lotus, evaluate their employees for their annual job performance review partly on how well they contribute their knowledge to the organization's knowledge repositories and how well they use and apply the knowledge that exists in these repositories. Scanty knowledge repository is a known problem in knowledge management systems (KMS) hence incorporating external knowledge sources for specific purposes within the repository is reasonable. In this paper, we propose a framework to capture unstructured and semi-structured data for learning new knowledge patterns and discuss how the result should be integrated into a knowledge management system or business intelligence dashboard. The relevance of this framework is underscored by the fact that discussion groups, social media, and online forums are significant modes of social interaction, hence organizations cannot continue to ignore the knowledge that could be extracted from these group-interactive platforms that generate mostly unstructured data. Furthermore, researchers through the years have noted the importance of the knowledge that emanate from social interactions and social platforms and the need to harness such knowledge. [5] noted that theories of organizational learning do not address the critical notion of externalization, and organizational learning has paid little attention to the importance of socialization. The socialization mode refers to the conversion of tacit knowledge to new tacit knowledge through social interactions and shared experience among organizational members [6]. In other words, knowledge can be generated through social interactions. Verma and Singh [7] highlighted that integrating relevant information from diverse sources and utilizing it all for decision-making purposes is still a huge challenge. Informal and formal channels, such as the intranet or corporate portals, should be employed to help access knowledge [8]. We respond to this challenge raised by [7] by creating a framework that allows unstructured data from diverse sources to be processed, and insights generated from the data, stored as organizational knowledge or used for organizational decision-making. In this paper, we make a case that;

- A new paradigm of organizational knowledge should not only leverage knowledge internally within the organization but also externally among reviewers, customers, users, consumers, and other stakeholders.
- Unstructured data from non-traditional sources such as social-interactive platforms should be extracted and harnessed.
- Unstructured data that has been processed and analyzed should be stored on a knowledge management system or monitored as graphical outputs on business intelligence dashboards for interesting trends and patterns.

The rest of the paper is structured as follows; section 2 will examine the frameworks available in the literature for capturing unstructured and semi-structured knowledge and data. In section 3, we will present our framework for capturing and processing unstructured and semi-structured data. Section 4 presents the evaluation of the framework and we conclude the paper in section 5.

2. LITERATURE OVERVIEW

Due to the inextricable link between data and knowledge according to [9], and [10], this section of the literature review will focus on published frameworks for analyzing unstructured and semi-structured data and knowledge.

Orenga-Rogla in [11] developed a knowledge management 2.0 development framework made up of content module, transfer module, enrichment module, and decision-making module. The

content module depicts are knowledge worker with tacit knowledge in the form of experience, thoughts, and competence. The transfer module components show how the knowledge worker can externalize the tacit knowledge via web 2.0 tools such as social networks, blog, and forums. The enrichment module focuses on using natural language processing and data mining to harness the knowledge while the decision-making module uses the harnessed knowledge to make decisions. While we agree with the authors that harnessing knowledge to be stored in a knowledge management systems should incorporate machine learning and data science tools which the authors used in the enrichment module, we note that the framework does not make the distinction that tools required to enrich unstructured data vary according to the nature of the data. Verma and Singh [7] developed a methodology for the integration of multi-structured data that emphasizes on generating superimposed data visualizations to facilitate interactive data exploration. The framework developed by the researchers is focused on event-driven analysis based on data captured from the web and other sources. The emphasis has been towards the retrieval of all events around a time-frame to ensure that a business analyst has a complete view of all events that happened during the period. The solution is suitable for dynamic businesses that are driven by events and quick changes in the landscape such as journalism or product marketing and not necessarily suitable for processing and storing knowledge in a knowledge management system.

In [12], Dey and others proposed a solution to integrate unstructured and structured data into enterprise analytics. In their approach, structured data was treated in the form of a time series that capture enterprise performance information such as weekly progress reports, sales figures, revenue, and stock prices, while unstructured data was taken from customer reports, reviews and feedback, discussion forums, blogs, and social media. According to [12], the framework exploits text processing and mining techniques for information extraction from unstructured sources and allow for multiple heterogeneous inputs to automate the process of knowledge discovery through correlation of information components extracted from the data. We learned some valuable lessons for our framework from the work done by [12] however, our framework is different in that we treat the various types of unstructured data uniquely, recognizing that the techniques for harnessing and processing the data will vary according to the type, a distinction they did not make in the paper.

Cheung and his colleagues in [13] developed a framework for the elicitation of knowledge from unstructured information. Their solution called multi-faceted and automatic knowledge elicitation system (MAKES) integrates the processes of collecting data, classifying unstructured information, modeling knowledge flow and social network analysis, and makes all of these actions into a connected process to audit unstructured information [14]. The system allows for retrieving, automatic classification, capturing and sharing of knowledge from unstructured information from emails, office documents, forums, bulletin boards, and blogs, which would contain multiple concepts that could be abstracted at different levels. The researchers were able to demonstrate the viability of their solution through a trial implementation and verification test conducted in the electronics industry.

3. PROPOSED FRAMEWORK

We propose a framework for capturing unstructured and semi-structured data and processing them to create outputs that are capable of providing insight and new knowledge patterns for the organization. Our method is supported by [15] who said that the development of the artifact should be a search process that draws from existing knowledge. Other frameworks tend to focus on one type of format, especially textual data. To the best of our knowledge, our framework is the first to incorporate all four types of unstructured data (text, image, audio, and video) in one framework.

3.1. Data Acquisition

The application-programming interface (API) on the framework provides data from different sources such as discussion groups, online forums, customer reviews and reports, social media, telephone conversation, mobile phone marketing, video conferencing, etc. The type of data will determine the techniques. If for instance, we extract textual data from blogs, discussion groups, and customer reviews, for preprocessing, we would apply metadata extraction, de-duplication, tagging, stemming, filtering, and stop word removal including punctuation marks and numbers. We will now walk through the techniques in the framework using textual data analytics.

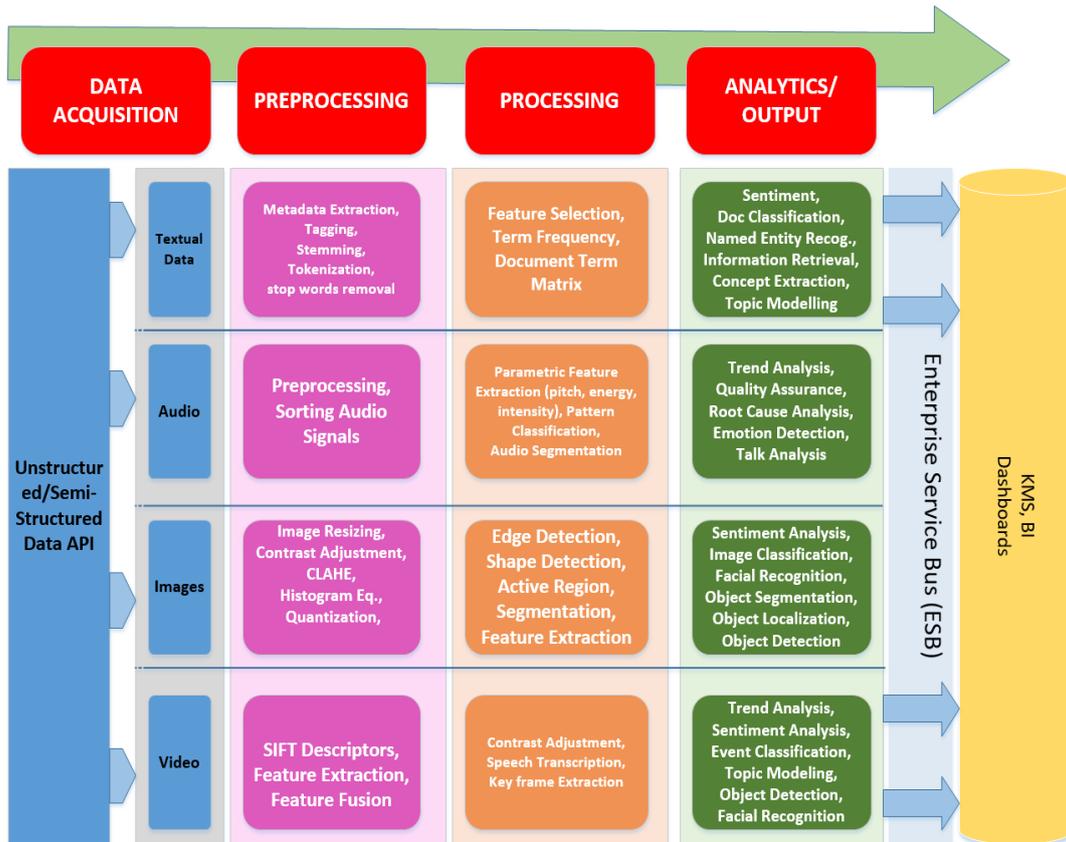


Figure 1. Proposed Framework for Capturing Unstructured and Semi-Structured Data

3.2. Pre-processing

De-duplication, also known as deduping, is a technique for removing duplicate copies and repeating textual data from the corpus. It has the benefit of reducing preprocessing and processing overhead. Textual data that contains lots of duplicates could lead to misleading output hence, it is important to apply this technique at the early stages of preprocessing.

Metadata Extraction: Unique properties of textual data such as the author, date, and time of creation, title, and subject may be extracted to provide relevant context during the preprocessing stage.

Tagging: We can apply named entity recognition (NER) or parts of speech (POS) tagging depending on the desired output. NER would allow for the identification of person names,

organizations, locations, times statement, etc., that is present in the data while POS tagging will allow for classifying words and labeling them accordingly.

Tokenization: Given the textual data under preprocessing, we apply the tokenization technique to split up the text into individual words, phrases, or whole sentences.

Stemming returns every derived word to its root or base form. This densifies the data, reduces the number of words used in the corpus, and results in more efficient processing of corpus.

Lemmatization procedure works similarly to stemming; however, it uses a dictionary to ensure that the derived word returns to the dictionary or base form.

Stop Words: Prepositional words, conjunctions, and commonly used words are removed from the corpus using the stop words removal procedure. So also, irrelevant numbers such as page numbers and punctuation marks.

3.3. Processing

Feature Selection technique allows us to select a smaller group or subset of terms in the corpus and using only this subset in further processing and analysis. This increases the efficiency of the algorithm by decreasing the vocabulary lexicon.

Term Frequency calculation gives the number of occurrences of a given term in the textual document. There are various term frequency calculations we could use but it depends on the objectives of the analytics. Examples include simple term frequency $tf(\text{term}, \text{document})$, inverse document frequency (idf), combined term frequency-inverse document frequency (tf-idf), term frequency adjusted for total document length (term count/total term count in document) and augmented frequency (term frequency/highest term frequency in document).

Document Term Matrix (DTM) calculates the term frequency per document and creates an (m by n) matrix where m is each document in the corpus and n is the terms in the document.

3.4. Analytics/Output

Sentiment Analysis: Sentiment analysis could be applied to textual data from sources such as discussion groups, online forums, customer reviews, and reports and social media to determine the overall attitude of users, customers, etc. to a product or service that an organization creates. The knowledge derived from the analysis could provide an early warning or inform the strategy for new product development.

Concept Extraction technique results in the extraction of concepts from the text. A common technique for concept extraction is the use of Word Tree. When certain terms occur side by side very frequently, there is a probability that it may be alluding to a concept.

Classification techniques will be used to manage, sort, and group textual data in predefined categories to increase information discovery and make all discovered knowledge available and useable to support decision-making.

Document Classification is a technique that is used to determine the major subject or theme of a document and then assign the document to predetermined classes or categories. Document classification is a useful technique in library science and information science for document management.

Information Retrieval is a technique that is used on the web, knowledge management systems and other types of information systems to ensure that the right information is retrieved in the form of a search result.

Topic modeling is used to find hidden or abstract topics that are embedded in a collection of textual documents.

3.5. Integration with Traditional KMS and Business Intelligence (BI) Dashboards

We propose an Enterprise Service Bus (ESB) as the middleware for integrating analytics output with the knowledge management system.

An Enterprise Service Bus (ESB) combines event-driven and service-oriented approaches to simplify the integration of business units, bridging heterogeneous platforms, and environments [16]. An ESB is an ideal backbone for implementing service-oriented architectures because it provides a universal mechanism to interconnect all the services fully integrated business solutions without compromising security, reliability, performance, and scalability [17]. It supports synchronous and asynchronous, facilitating interactions between one or many stakeholders - one-to-one or many-to-many communications [16]. Industry-standard enterprise service bus solutions include IBM WebSphere, Oracle ESB, and SAP PI. We argue that insights and new knowledge gained from analyzing these data should be integrated into a traditional knowledge management system to enhance the richness of the knowledge available in the KMS to improve organizational performance and decision-making. Organizations often use business intelligence dashboards to monitor key performance indicators and other relevant metrics drive the business. A dashboard provides a rich user interface that displays the information in a graphical form using a variety of elements including charts, tables, and gauges. These elements reduce the time spent on analyzing the data using databases and thus assist in automating the business decision-making process [18]. While analytics results should be stored in a KMS, we argue that relevant analytics that is used in decision making should be placed on business intelligence dashboards for continuous monitoring. As such we provide our evaluation results in the form of graphics that could also be displayed in business intelligence dashboards.

4. FRAMEWORK EVALUATION

The effectiveness of an artifact such as a framework should be determined using an established design science evaluation method. [19] and [20] list the proven methods for evaluating a design science artifact. We prefer the more recent [20] that lists evaluation method types to include Logical Argument, Expert Evaluation, Technical Experiment, Subject-based Experiment, Action Research, Prototype, Case Study, and Illustrative Scenario. It defines Illustrative Scenario as the application of the artifact to a synthetic or real-world situation aimed at illustrating the suitability or utility of the artifact. We use the Illustrative Scenario as our method of evaluating the framework because it is the most suitable method given the nature of the artifact and the time available to complete the evaluation. We acknowledge that Case Study and Action Research evaluation methods may be more robust methods to use but they would take considerably more time to complete.

To validate the framework, we use Python and Beautiful Soup library as API to crawl textual data. An Application Programming Interface (API) has been defined as a specification that defines an interface for Software components to communicate with each other [21]. In this scenario, the specification is the HTML protocol and the software components are the website and python IDE. As highlighted earlier, each of the separate types of unstructured data in our

framework may require separate procedures for evaluation. [19] highlighted that the evaluation of a designed IT artifact requires the definition of appropriate metrics and possibly the gathering and analysis of appropriate data. As such, we focus on textual data for our evaluation. We choose www.studentsreview.com and we scrap ten students' reviews of the institution where this research was performed. We name the students student_1 to student_10.

We write our API programming code as follows:

```
def api(url):
    '''Returns student review from studentreviews.com.'''
    page = requests.get(url)
    sp = BeautifulSoup(page.text, 'html.parser')
    text = [p.text for p in sp.find(class_="portfolioContainer")]
    return text
```

4.1. Pre-processing

For our purposes, we will remove punctuations, brackets, parentheses, and common English stop words. We do not need to apply the stemming processing.

```
def clean_corpus(text)
    text = text.lower()
    text = re.sub('im', '', text)
    text = re.sub('ive', '', text)
    text = re.sub('got', '', text)
    text = re.sub('isnt', '', text)
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

4.2. Processing

The purpose of the text analytics determines the type of processing we apply to the text corpus. Based on our framework, text analytics can be performed for text classification, concept extraction, sentiment analysis, topic modeling, named entity recognition, event extraction, and information retrieval. For our evaluation, we apply document-term matrix to allow us to perform word frequency analysis for each student. We build a word cloud and show the level of profane words used by each student. The plot for each processing step is shown in the figures below:

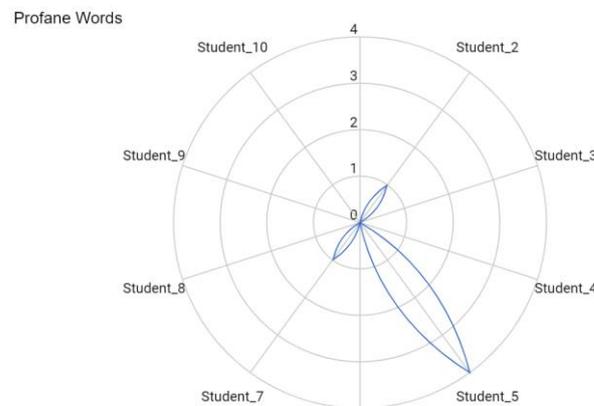


Figure 2. Profane Words Radar Chart

In the profane word radar chart, we show the level of profane words that each student is using. In our code description, profane words include words such as hell, f*ck, assh*le. The profane word analysis reveals that student_5 uses the most profane words with a total of four words. Student_2 and student_7 used one profane word each in their review while the rest of the students did not use profane words.

We show a word cloud to see the words that the students used most frequently in the reviews. From the word cloud, we see that some students focused their reviews on the school as a whole, while others on their major, school clubs, social activities, and even kids.

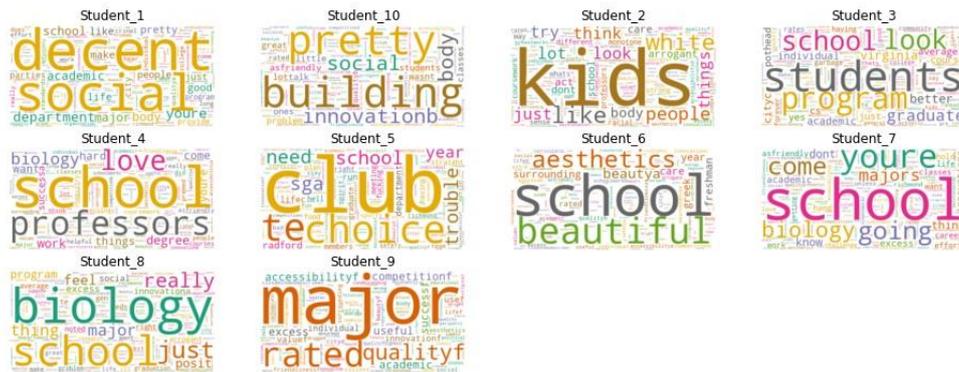


Figure 3. Word Cloud showing most frequent words for the students

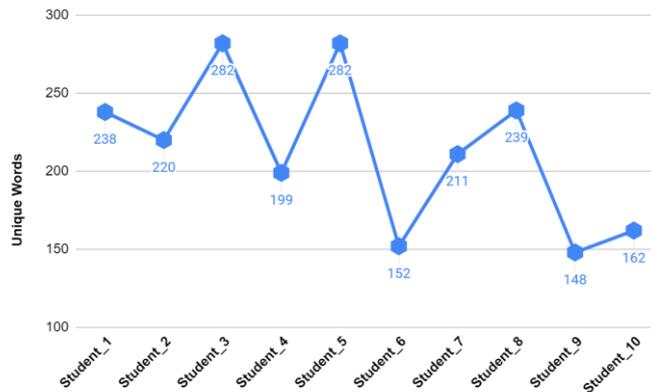


Figure 4. Unique Words of each student

The unique words plot shows the number of unique words a student used that none of the other students in the review used. The unique word plot is an indication of the length and detail of the review. It is easy to run a simple word count that would tell the length of the review however, unique words are better in this scenario because some words such as faculty, course, class, classroom, campus, school, etc. will occur across all the reviews, hence unique word count is a better indicator of how lengthy and detailed the review is. Unique word count is also an indicator of the unique points the student makes in the review. A review with a high number of unique words is expressing more views and touching on more points than one with a lesser number of unique words. The result showed that student_3 and student_5 had more to say that is different from what every other person is saying while student_9 has the least to say that is different from what others have said.

4.3. Analytics Type

The analytics type that fits the most based on the text corpus is sentiment analysis. Sentiment analysis usually relies on applying machine learning techniques to classify texts based on a collection of features extracted from the text using Natural Language Processing techniques, such as the presence of certain words or the coverage of some topics [22]. Topic modeling does not fit the corpus because we already know what the topic is – student giving their opinion and experience about the university they graduated from. Other types of text analytics such as concept extraction, document classification, and named entity recognition also are not the best fit given our context and nature of corpus.

The sentiment analysis plot is shown in figure 5. The horizontal axis represents the polarity of the sentiment while the vertical axis represents the subjectivity or opinion level of the review. For the polarity, zero represents a generally neutral review, values less than zero represent a negative review overall, and values greater than zero represent a positive review overall. An objective sentence expresses some factual information about something, while an opinion sentence expresses some personal opinions, beliefs, feelings, allegations, desires, suspicions, and speculations. Objectivity and opinion are opposites hence a review that is high in opinion is automatically low in objectivity and vice versa. Student_7 scored the least in opinion which on the other hand means that student_7 presented the most objective review. Student_10 presented the most opinionated review which in turn means that student_10 review is the least objective. The polarity of our analysis ranges from -0.03 for student_2 to +0.21 for student_6. Student_6 submitted the most positive review about the school and student_2 submitted the most negative review.

The sentiment analysis time series for all 10 students are shown in figure 6. The vertical axis represents the polarity of the review and the horizontal axis is the timeline from 0 to 10, with zero representing the starting sentence in the review and ten the last sentence. The orange horizontal line is the neutrality line. This plot shows how the sentiment of the students varies across the entire length of the review. Some students stayed entirely positive in their review such as student_4 and student_6 which is an indication that they felt entirely positive about their experience at this university. Most other reviewers were swinging from positive to negative sentiments with student_2 having an overall negative polarity.

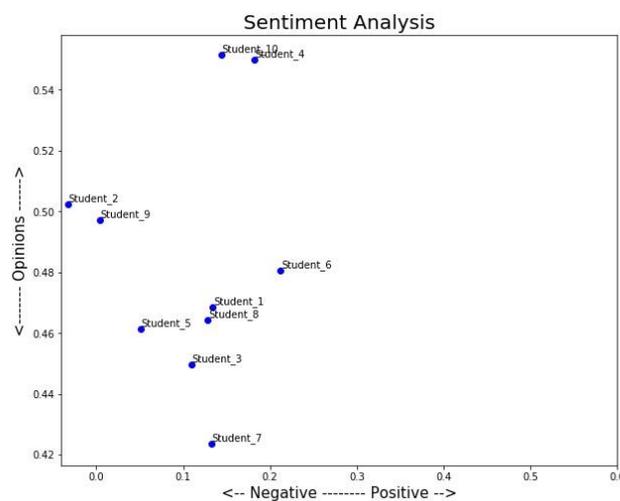


Figure 5. Sentiment Analysis

4.4. Integration to a Knowledge Management System

The sentiment analysis plot can be stored in an organizational knowledge management system and reviewed periodically based on the stipulations of the organizational policy. The graph analysis could also be integrated into a business intelligence dashboard and reviews with extreme negative or positive polarity could be further investigated to find what the student liked the most or hated the most about their experience. Ideas and patterns that are repeating across multiple students could inform future decisions to improve college experience for students.

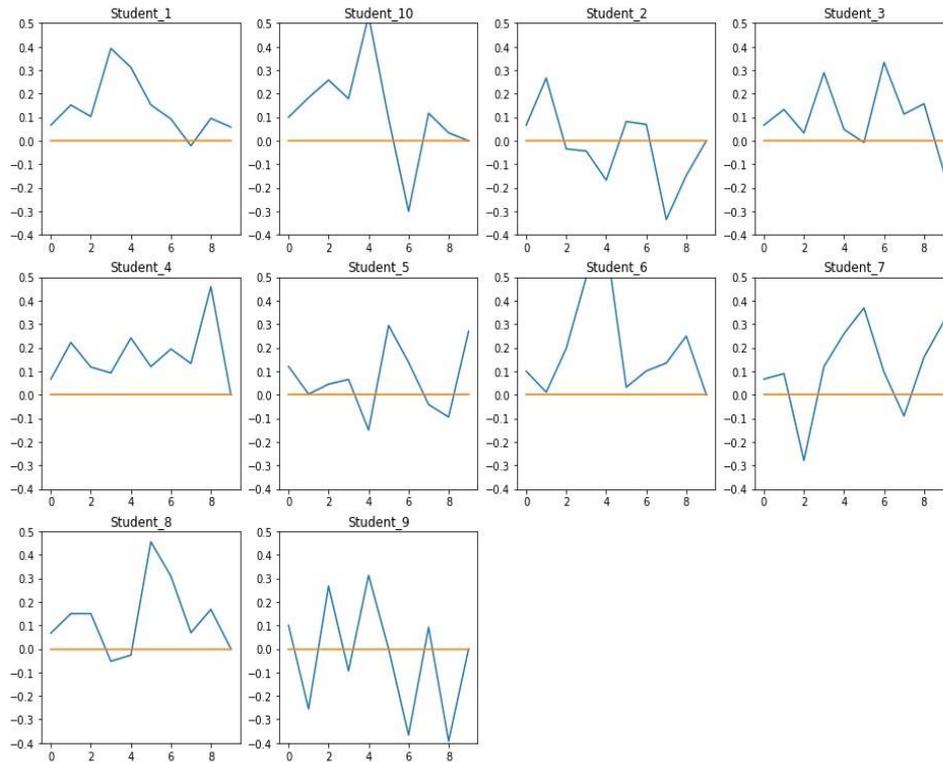


Figure 6. Sentiment Analysis Time Series for each student

5. CONCLUSIONS

In this paper, we have developed a framework to process and unstructured data and knowledge from multiple sources and incorporate the process output into a traditional knowledge management system. We note that the framework would be infeasible to evaluate by one straightforward procedure, but rather would require that specific instantiation be made from the framework and evaluated. In evaluating the framework, we leaned on the design science research evaluation methods provided by [20] and we used the Illustrative Scenario method. An instance of the framework was created that used student reviews from studentsreview.com to perform sentiment analysis. We used python and beautiful soup as API to crawl student review data from the website and we performed unique word count, profane word count, word cloud, sentiment analysis, and sentiment analysis time series. We propose that the result from the sentiment analysis and sentiment analysis time series could be stored in a knowledge management system and reviewed from time to time or could form an input to a business intelligence visualization tool for real-time monitoring. For future research, we hope to develop a software application interface that can sentiment-analyze user reviews from several social network feeds.

REFERENCES

- [1] B. Ramesh and A. Tiwana, "Supporting collaborative process knowledge management in new product development teams," *Decision support systems*, vol. 27, no. 1–2, pp. 213–235, 1999.
- [2] A. L. Leal-Rodríguez, J. L. Roldán, A. G. Leal, and J. Ortega-Gutiérrez, "Knowledge management, relational learning, and the effectiveness of innovation outcomes," *The Service Industries Journal*, vol. 33, no. 13–14, pp. 1294–1311, 2013.
- [3] Z. Soltani, B. Zareie, L. Rajabiun, and M. F. A. Agha, "The effect of knowledge management, e-learning systems and organizational learning on organizational intelligence," *Kybernetes*, vol. ahead-of-print, no. ahead-of-print, Jan. 2019, doi: 10.1108/K-12-2018-0672.
- [4] J. Liebowitz, *Building Organizational Intelligence: A Knowledge Management Primer*. CRC Press, 2019.
- [5] I. Nonaka, "A dynamic theory of organizational knowledge creation," *Organization science*, vol. 5, no. 1, pp. 14–37, 1994.
- [6] I. Nonaka and R. Toyama, "The theory of the knowledge-creating firm: subjectivity, objectivity and synthesis," *Industrial and corporate change*, vol. 14, no. 3, pp. 419–436, 2005.
- [7] I. Verma and L. Singh, "Multi-structured Data Analytics using Interactive Visualization to aid Business Decision Making," in *Proceedings of the 10th Annual ACM India Compute Conference*, 2017, pp. 21–29.
- [8] M. Corso, A. Martini, L. Pellegrini, S. Massa, and S. Testa, "Managing dispersed workers: the new challenge in Knowledge Management," *Technovation*, vol. 26, no. 5, pp. 583–594, May 2006, doi: 10.1016/j.technovation.2005.02.003.
- [9] T. H. Davenport, P. D. of M. I. S. of M. T. H. Davenport, and L. Prusak, *Working Knowledge: How Organizations Manage what They Know*. Harvard Business Press, 1998.
- [10] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995.
- [11] S. Orega-Roglá and R. Chalmeta, "Methodology for the Implementation of Knowledge Management Systems 2.0," *Bus Inf Syst Eng*, vol. 61, no. 2, pp. 195–213, Apr. 2019, doi: 10.1007/s12599-017-0513-1.
- [12] L. Dey, I. Verma, A. Khurdiya, and S. Bharadwaja, "A framework to integrate unstructured and structured data for enterprise analytics," in *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 1988–1995.
- [13] C. F. Cheung, W. B. Lee, W. M. Wang, Y. Wang, and W. M. Yeung, "A multi-faceted and automatic knowledge elicitation system (MAKES) for managing unstructured information," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5245–5258, 2011.
- [14] C. F. Cheung, W. B. Lee, W. M. Wang, Y. Wang, and W. M. Yeung, "A multi-faceted and automatic knowledge elicitation system (MAKES) for managing unstructured information," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5245–5258, May 2011, doi: 10.1016/j.eswa.2010.10.033.
- [15] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.
- [16] J.-L. Marechaux and I. Corporation, "Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus," p. 8, 2006.
- [17] M. Falko, "Enterprise Service Bus," in *FREE AND OPEN SOURCE SOFTWARE CONFERENCE*, 2007, Accessed: Apr. 23, 2019. [Online]. Available: https://programm.froscon.org/2007/attachments/15-falko_menge_-_enterprise_service_bus.pdf.
- [18] B. Hansoti, "Business Intelligence Dashboard in Decision Making," p. 61, 2010.
- [19] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.
- [20] K. Peffers, M. Rothenberger, T. Tuunanen, and R. Vaezi, "Design Science Research Evaluation," in *Design Science Research in Information Systems. Advances in Theory and Practice*, Berlin, Heidelberg, 2012, pp. 398–410, doi: 10.1007/978-3-642-29863-9_29.
- [21] P. Mahiddini, "Dynamic application programming interface publication for providing web services," *US9652314B2*, May 16, 2017.
- [22] K. Henrikson, F. Rodrigues, and F. C. Pereira, "Chapter 5 - Data Preparation," in *Mobility Patterns, Big Data and Transport Analytics*, C. Antoniou, L. Dimitriou, and F. Pereira, Eds. Elsevier, 2019, pp. 73–106.

AUTHORS

Gerald Onwujekwe is a Ph.D. candidate in Information Systems at Virginia Commonwealth University, Virginia, USA. His research interests include Graph Databases, Convolutional Neural Networks, and Deep Learning and Text mining.



Kweku-Muata Osei-Bryson is a professor of Information Systems at Virginia Commonwealth University and Research Fellow of the Information Systems Research Institute since Fall 1998. Previously I was Professor of Information Systems and Decision Analysis in the School of Business at Howard University, Washington, DC, U.S.A. I have also worked as an Information Systems practitioner in both industry and government. Analytics & Data Science, Data Mining, Cyber-Security, Knowledge Management, ICT for Development, Expert & Decision Support Systems, ICT & Productivity, Multi-Criteria Decision Analysis.



Nnatubemugo Ngwum is a doctoral candidate at Towson University, Maryland, USA. Obtaining a master's degree from the University of Manchester (UK) in computer security, his research interests extend beyond information systems and their security to include Internet of Things (IoT), Internet of Things security, and Human-Computer Interaction.



FACIAL EXPRESSION RECOGNITION USING COMBINED PRE-TRAINED CONVNETS

Raid Saabni^{1, 2} and Alon Schclar¹

¹School of Computer Science,
The Academic College of Tel-Aviv Yaffo, Tel-Aviv, Israel
²Traiangle R&D Center, Kafr Qarea, Israel

ABSTRACT

Automatic Facial Expression Recognition (AFER), has been an active research area in the past three decades. Research and development in this area have become continually active due to its wide range of potential applications in many fields. Recent research in the field presents impressive results when using Convolution Neural Network (CNN's, ConvNets). In general, ConvNets proved to be a very common and promising choice for many computer vision tasks including AFER. Motivated by this fact, we parallelly combine modified versions of three ConvNets to generate an Automated Facial Expression Recognition system. This research aims to present a robust architecture and better learning process for a deep ConvNet. Adding four additional layers to the combination of the basic models assembles the net to one large ConvNet and enables the sophisticated boosting of the basic models. The main contribution of this work comes out of this special architecture and the use of a two-phase training process that enables better learning. The new system we present is trained to detect universal facial expressions of seven\eight basic emotions when targeting the FER2013 and FER2013+ benchmarks, respectively. The presented approach improves the results of the used architectures by 4% using the FER2013 and 2% using FER2013+ data sets. The second round of training the presented system increases the accuracy of some of the basic models by close to 3% while improving the accuracy of the whole net.

KEYWORDS

Automatic Facial Expression Recognition, Convolutional Neural Networks, Machine Learning, Boosting, Deep Learning.

1. INTRODUCTION

Psychologists found that verbal and vocal parts of a message contribute only 41% of its meaning while facial movements and expressions contribute 55% of the effect of that message. This fact means that the facial part does the major contribution to human communication and interaction [17]. Therefore, developing Automatic FER applications would be widely applicable for many real-world tasks, which can get the significant benefit of reliable systems that automatically recognize facial expressions and emotions. Some of such fields, are Human-Computer Interface, Human Emotion Analysis, Image Retrieval, User Profiling, Medical Care and Cure, Video Games, Neuro Marketing, and many more. People can vary significantly in the way they show their expressions for even the same person and expression, which makes AFER a more challenging problem. Images also can vary in brightness, background, and pose, and these variations are emphasized when considering different persons with variations in shape, ethnicity, and other factors.

Facial expression recognition is a task naturally done by humans daily, but it is a complex task for computer programs. This problem is challenging for computers because it is very hard to extract and classify expression's features when images may vary a lot not only in the way that the subjects show their expression but also due to different conditions of lighting, brightness, position, and background. Other difficulties may include face position and direction, face partial occluding by objects in the scene, or due to bad light conditions causing high variations of illumination, which may easily lead to losing main features of facial expressions.

The work done in the 1970s by the psychologist Paul Ekman [20] and his colleagues, is an important milestone in the study of facial expressions and human emotions. This important work has significant importance and a large influence on the development of modern-day automatic facial expression recognizers. This work leads to adapting and developing the comprehensive Facial Action Coding System (FACS), which has since then become the standard for facial expression recognition research. Facial expressions are extremely important in any human interaction, and additional to emotions, it also reflects on other mental activities, social interaction, and physiological signals. Ekman et. al. identified six facial expressions that are universal across all cultures: Anger, Disgust, Fear, Happiness, Sadness, and Surprise. These six emotions in addition to the natural one, are still used for most of the modern automatic facial expression research. State of the art research on competitions and challenges in the field such as Emotion Recognition in the Wild (*EmotiW*) and Kaggle's Facial Expression Recognition Challenge *FER2013* use these seven emotions in their competitions.

In traditional approaches, facial expression recognition usually consists of three main steps. In the first step, the system detects the face region from an image or sequence of images. This is mostly followed by a pre-processing step to emphasize the relevant features and neglect the irrelevant data. In the next step, features are extracted from the region of interest. Selecting a compact and effective facial representation and features from the face image is a vital step for successful facial expression recognition. The last step uses the extracted features to train and obtain a classifier. Many of the recent systems targeting *AFER* are based on Convolutional Neural Networks (*ConvNet*), using existing and new variations of ConvNet architectures. These approaches present many of the state-of-the-art results in tasks of object classification including facial expression recognition. Unlike traditional approaches, in many cases, no human crafted, and designed features are needed, and the system operates as a start-to-end technique.

In the proposed system, we have designed a start-to-end system based on *ConvNets*. Motivated by our previous work [24], we have used a pre-processing step to extract the region of interest (*ROI*), which has already proved to improve results. We have also used normalization and data augmentation as an additional practice generally used to improve generalization ability [27,29,25]. In section 3, we review some of the previous works in the related field. In section 4, we present an overview of our approach. Detailed experimental results of the proposed system are presented in section 5 and concluding and future work directions are presented in section 6.

2. PREVIOUS WORK

Though much progress has been made, automatic recognizing of facial expressions with high accuracy remains difficult due to the complexity and variability of facial expressions. Generally, in classic approaches, the system includes two stages: feature learning and selection, and classifier construction. In the first stage, features are extracted from either static images or video sequences of images, to characterize facial appearance/geometry changes caused by activation of target expression. There are two common approaches to extract facial features: geometric and appearance feature-based methods. The geometric features measure the displacements of certain parts of the face such as eyebrows, eyes, mouth lines, and corners. This is based on the

assumption that expressions affect the relative position and size of various features, and that by measuring the movement and relative position of certain facial points, we can determine the underlying facial expression. In such a case, finding and tracking a crucial point in the face region is an important task of geometric feature measurement and face region analysis.

The idea of appearance-based methods assumes that emotions cause changes to face textures, such as wrinkles, bulges, forehead, regions surrounding the mouth, and eyes when performing a particular action. In appearance-based methods, image filters are applied to regions of interest, which can be any specific region in a face image, to extract feature vectors. These methods include Principal Component Analysis (PCA), Locality Preserving Projections (LPP), Linear Discriminate Analysis (LDA), Gabor wavelets, Local Binary Pattern (LBP), and others. As suggested by the psychological studies, the information specific areas such as around nose, eyes, and mouth are more critical for facial expression analysis. Therefore, a subset of features, which are the most effective to distinguish one expression from the others, are often selected to improve the recognition performance. In the second step, a classifier is obtained by training on a data set [10,14,18,19]. Recently, unsupervised feature learning approaches especially those based on Sparse-Coding [5,28,30] and Deep Learning Networks, have been employed to extract underlying features from facial images and have shown promising results in facial expression recognition and analysis.



Figure 1: (A) Samples of images from the (CK+) benchmark.
(B) Some samples of the different emotions from the JAFFE data set.

Convolutional Neural Networks are a category of Neural Networks that have proven very effective in image recognition and classification areas. *ConvNets* have been successful in identifying faces, objects, traffic signs, and many other computer vision tasks. *ConvNets* work better for image recognition and classification because they can automatically capture spatial features of the inputs due to their large number of filters. These features and filters are not hand-designed but are learned as a part of the training process. This fact makes Neural Networks in general and *ConvNets* specifically a better choice for start-to-end solutions for computer vision tasks. Yu and Zhang [29] achieved state-of-the-art results in 2015 on the EmotiW2015 data set, by using an ensemble of *ConvNets* having five convolutional layers each and using stochastic pooling rather than max pooling. They randomly perturb the input images to get an extra boost of 2-3% inaccuracy. They applied transformations to the input images at train time. At test time,

their model-generated predictions for multiple perturbations of each test example and voted on the class label to produce a final answer.

Kim et. al. [12] achieved a height test accuracy on EmotiW2015 by using an ensemble-based method with varying network architectures and parameters. They used a hierarchical decision tree and an exponential rule to combine decisions of different networks rather than simply using a simple weighted average to improve the results. They initialized weights by training networks on other FER data sets and using these weights for fine-tuning. Mollahosseini et. al. [1] have achieved the state-of-the-art results on FER2013 using a *ConvNet* which is consisted of two convolutional layers, max-pooling, and 4 Inception layers as introduced by *GoogLeNet*. The proposed architecture received a low-test accuracy of 47% when tested on the EmotiW2015 data set. Pramerdorfer and Kampel[22], review the state of the art in image-based facial expression recognition using *ConvNets*, and highlight algorithmic differences and their performance impact and by that identify existing bottlenecks and consequently directions for advancing this research field. Furthermore, they demonstrate that overcoming one of these bottlenecks leads to a performance increase. They used an ensemble of modern deep *ConvNets* to obtain a test accuracy of 75.2% on *FER2013*.

Saravanan et. al. [25], experimented with several different models, including decision trees and neural networks, and find that *ConvNets* work better for image recognition tasks since they can capture special features of the inputs due to their large number of filters. They propose a model consists of six convolutional layers, two max-pooling layers, and two fully connected layers. Upon turning off the various hyperparameters, this model achieved a final accuracy of 60%. In [9], the author reviewed the development of FER using VGGNet, ResNet, GoogleNet, and AlexNet tested on *FER2013*. After making some improvements based on the original methods of FER and training on the *FER2013* data set with different revised ways, the best result of accuracy they got is 64.24%.

Burkert et. al. [3], propose a convolutional neural network architecture for facial expression recognition. The proposed architecture is independent of any hand-crafted feature extraction and performs better than the earlier proposed convolutional neural network-based approaches. They tested their system on the standard datasets Extended Cohn-Kanade (CK+) and MMI to achieves an accuracy of 99.6% on (CK+) and 98:63% for MMI. For a comprehensive survey refer to [4,13,22,26]

3. OUR APPROACH

We have used three well-known *ConvNet* models already used for automatic facial expression recognition. Results were close to what has been reported in the literature and a minor improvement has been recorded when we slightly modified their architectures.

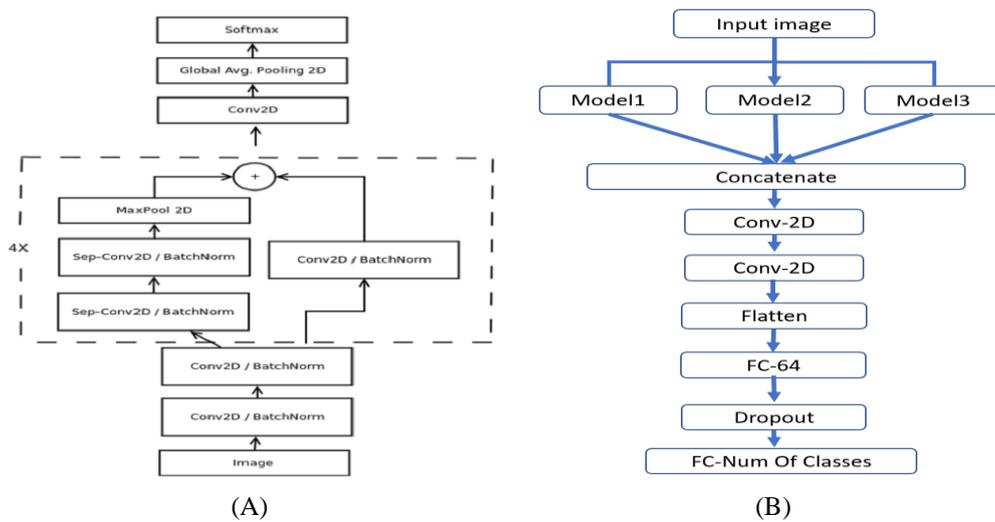


Figure 2: Model (A) is based on the architecture presented in [27] and Model (B) is our final ConvNet combining all the three pre-trained models and the four additional layers.

The novel contribution of the presented research came out when we used a combination of pre-trained versions of these three models to one large *ConvNet*. After combining these models in parallel, we added four additional layers and re-trained the net using two schemes. In the first, we have retrained the net while freezing the weights of the pre-trained layers, and at the second scheme, we retrained the whole net including the weights of the pre-trained layers.

The first scheme has been motivated by the idea of boosting or ensemble of classifiers using the four additional layers to learn the boosting parameters and lead to a 1%-2% improvement of accuracy. The second scheme enabled re-training the pre-trained models in addition to the whole net, and it improved the results by more than 4%. An interesting result of the re-training process came out when the updated re-trained weights of the basic models improved their results while training the whole net. All reported results were conducted on the *JAFFE* [16], (*CK+*) [15], *FER2013* [6], and *FER2013+* [7] data sets and will be presented in details at section 4.

3.1. Face Localization and Detection

Encouraged by results of previous works [9,24], we start facial expression analysis, by detecting and localizing the face in the given image. Locating the face within an image is termed as face detection and localization. As been reported in many papers the first and one of the best options to consider is the one developed in 2004 by Viola and Jones. The method is very fast and could rapidly detect frontal view faces by applying the AdaBoost learning algorithm on a simple class of features. The Authors, achieve excellent performance by using novel methods that could compute the features very quickly and then rapidly separate the background from the face [21].

The Viola-Jones algorithm uses five patterns to extract Haar-like features which are assumed to hold all the information needed to characterize a face. The number of the resulted Haar-like features is huge; therefore, the use of the integral image technique allows us to calculate them at a very low computational cost. To make sense of these features which can be seen mostly as weak classifiers, the Ada-boost [8] algorithm is used to generate a strong and accurate classifier based on a small set of the weak classifiers. Additional use of the Ada-boost method enables generating cascade classifiers, which produces a fast rejection mechanism of non-face areas efficiently. Many 24×24 images of faces are used to train and obtain a face detection algorithm

in real-world images quickly and efficiently. In the proposed system, we have used the Viola-Jones algorithm to detect the exact window of the main part of the face and then locate the eyes and mouth within that face. In the next step we have normalized the face image to include the face from the limits of the eyes horizontally, and eyes to mouth positions vertically. We crop the target window based on previous calculations from the image and re-size back to the original size.

3.2. Convolutional Neural Network Architectures and Pre-Trained Models

In this section, we present the three models we have used and combine to generate the proposed system. The input to these models is a $W \times H$ grey level images, where the width W and the height H are derived from the size of images used in the specific benchmark. In all models we have used Data Augmentation (*DA*), to generate more samples for the training set by applying transformations such as rotation, crop, shifts, shear, zoom, flip, reflection, and normalization. This is required usually to increase the size and the variability of the training set when it is not enough to learn data representations. We also have used $L2$ regularization of the weights to apply penalties on layer parameters during optimization which are incorporated in the loss function that the networks optimize. Batch normalization has been used after each layer to normalize the activation of the previous layer at each batch, to maintain the mean activation close to zero and the activation standard deviation close to one. This practice acts as a regularizer to handle the problem of internal shift co-variation.

The first model we have used is a deep ConvNet with very few free parameters when compared to other deep models in the area. The model presented in [27] is motivated by the idea of reducing the number of free parameters usually exist as weights in the final fully connected layers. This architecture combines the deletion of the final fully-connected layers and the inclusion of the combined depth-wise separable convolutions and residual modules, and by that could speed up the process of training. It uses Average Pooling having the same number of feature maps as the number of classes in the last convolution layer and the soft-max activation function, which enables completely removing the fully connected layers, see *Model(A)* in Figure 2. This architecture is a standard fully convolutional neural network composed of 9 convolution layers. Using Global Average Pooling reduces each feature map into a scalar value by taking the average over all elements in the feature map to force the network to extract global features from the input image. This architecture exchanges the 2-d convolutional layers with depth-wise separable convolutions which are composed of two different layers depth-wise convolutions and point-wise convolutions. This architecture succeeds to reduce the number of the weights within the *ConvNet* to approximately 600,000 parameters comparing to more than 140 million in the VGG-16 ConvNet and achieved an accuracy of 66% on the *FER-2013* data set [27].

VGG-16 is one of the state-of-the-art architectures for convolutional neural networks. This architecture consists of 16 weight layers that include 13 convolution layers followed by three fully connected layers. All layers use a 3×3 filter size of one pixel for stride and padding. The convolutional layers are divided into 5 groups and each group is followed by a max-pooling layer. In the VGG-16, each group includes a Convolutional layer with several filters and ends with a Max-pooling layer carried out over a 2×2 window with stride 2. The number of filters of each convolutional layer is multiplied by 2 when moving from one group to the next and starts from 64 in the first group and 512 in the last group, where all these features are of the size 3×3 . An additional group of three fully connected layers, which have most of the weights of the net comes after the first 5 groups. The first two layers in this group have 4096 nodes each, and the third contains seven/eight channels (one for each class), The first two layers use *ReLU* activation function, and the last one uses the *SoftMax* activation function for the final classification, See Model (C) in Figure 3 for the full architecture.

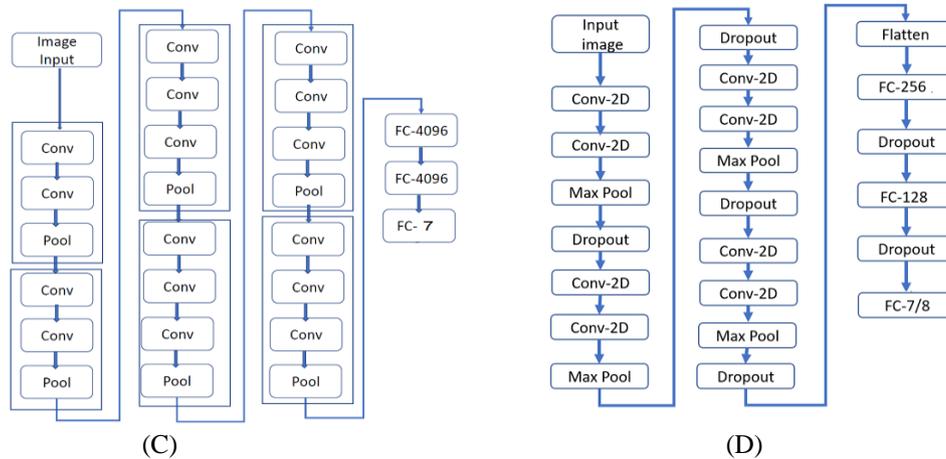


Figure 3: Model (C) is based on the VGG16 ConvNet Architecture and Model (D) is our modified version of VGG16 with less free parameters.

The third system was inspired by the VGG architecture and consists of 11 layers with close to 1,460,000 free trainable parameters. It has 4 Groups of two Conv-2D layers followed by the max-pooling layer. The number of filters of the convolutional layers in each group is multiplied by 2 when moving from one group to the next and starts from 32 in the first one, leading to that the last group reaches 256 filters of 3×3 size. After the fourth group comes a group of three fully connected layers which had most of the weights of the net where the first two have 256 and 128 nodes, and the third contains seven/eight channels (one for each class) using *ReLU* and *Softmax* activation functions for the final classification, See Model (D) in Figure 3 for the full architecture.

The full system uses the three models as basic building components and combines them to one large convolutional neural network. The output of each model is a flatten layer with N nodes derived from the number of classes (N is the number of classes which is 7 and 8 for FER and FER+ data sets receptively). The output layers of the three models having the same size are concatenated to a three channels layer. Two Convolutional layers follow the concatenation layer and the final two layers are fully connected layers with 64 and N nodes respectively, See Model (B) in Figure 2 for the full architecture.

4. DATA SETS AND EXPERIMENTAL RESULTS

To evaluate our system, we have used four standard benchmarks, the (JAFEE), the (CK+), and two versions of the FER2013 benchmark. Because of the inaccurate labeling problem of the (FER2013) images reported in [29] and other papers, we have used the re-tagged version of (FER2013) named (FER2013+) [7,29], see Figure 4 for examples of such labeling errors. The JAFFE [16] data set contains 213 images of Japanese females, collected by Komachi and Yoba at Kyushu University, Japan. Ten subjects were asked to pose several different facial expressions, where pictures were taken, through remote control while looking towards the camera. Original images have been rescaled and cropped such that the eyes are roughly at the same position with a resolution of 256×256 pixels. The number of images corresponding to each of the 7 categories of expression is roughly the same, few of them are shown in Figure 1 (B). The second benchmark is the (CK+) database [15] which contains labeled image sequences for 123 subjects, where each sequence has one of 6 expressions (contempt replaced disgust, natural is the first image in each sequence), see Figure 1 (A).

For each image sequence, only the last frame (the peak frame) is provided with an expression label. Some of the subjects did not have all the 6 expressions, so the final number of subjects we have used for this case was only the 10 subjects from whom we have all the 6 expression categories. Three images from the last five frames were extracted for training/testing purposes from each image sequence in order to expand the data set. Two images of the first two frames from each sequence were extracted and labeled as neutral expressions. To evaluate the presented approach, we have generated data sets for JAFFE and CK+ benchmarks taking out all images of all subjects. In the next step, we have divided them randomly into three sets which we have used for training, validation, and testing.



Figure 4: Samples of images from the FER2013 benchmark with re-tagged labels. The upper labelling is the original FER2013 labelling and the lower is the FER2013+ labelling.

The *FER2013* data set [6], is provided by the Kaggle community website and consists of about 37000 greyscale images of faces with the size 48×48 . The images are pre-processed and registered so that the face is centred and occupies about the same amount of space in each image. Each image is categorized into one of the seven classes that express different facial emotions. The data set is divided into three different sets with the sizes, 29000, 4000, and 4000 images for training, validation, and testing, respectively. The images in the original FER data set was filtered and labeled by human labelers with emotion-related keywords, but the label accuracy is not very high [11], see a few such examples in Figure 4. The authors in [7,29], re-tag the *FER2013* data set using crowdsourcing. For each input image, they asked crowd taggers to label the image into one of 8 classes while adding the 'contempt' emotion as an additional one to the existing 7 classes. The taggers are required to choose one single emotion for each image and the gold standard method has been adopted to ensure the tagging quality.

Table 1: The confusion matrix with accuracy rates for each emotion category using the CK+.

Results of the CK+ data set							
Class	Neutral	Surprise	Anger	Disgust	Fear	Happiness	Sadness
Neutral	98.8 %	0.0 %	0.2 %	0.6 %	0.2 %	0.2 %	0.0 %
Surprise	0.0 %	98.5 %	0.7 %	0.0 %	0.5 %	0.3 %	0.0 %
Anger	0.9 %	0.0 %	95.5 %	0.6 %	0.9 %	0.6 %	1.5 %
Disgust	0.0 %	0.0 %	1.7 %	97.8 %	0.0 %	0.3 %	0.2 %
Fear	0.3 %	0.6 %	1.6 %	2.2 %	94.3 %	0.4 %	0.6 %
Happiness	0.1 %	1.1 %	0.0 %	0.0 %	0.7 %	98.1 %	0.0 %
Sadness	0.1 %	0.1 %	2.2 %	0.0 %	3.5 %	0.0 %	94.1 %
Average				96.81 %			

Ten taggers were asked to label each image, thus obtaining a distribution of emotions for each face image. They randomly chose 10000 images from the data set and assume that the majority of the 10 labels are a good approximation to the ground truth labels. When they have fewer taggers, they compute how many of the majority agree with the ground truth emotion and show that when there are 3 taggers, the agreement is merely 46%. With 5 taggers, the accuracy improves to about 67% and, with 7 taggers, the agreement improves to above 80%. they concluded that the number of taggers has a high impact on the final label quality [29]. With 10 annotators for each face image, they generate a probability distribution of emotion capture by the facial expression, which enables experiment to be held with multiple schemes during training (Categorical and Probability).

Table 2: The confusion matrix with accuracy rates for each emotion category using JAFFE.

Results of the JAFFE data set							
	Neutral	Surprise	Anger	Disgust	Fear	Happiness	Sadness
Neutral	98.5 %	0.0 %	0.2 %	0.8 %	0.3 %	0.2 %	0.0
Surprise	0.2 %	98.2 %	0.4 %	0.0 %	0.6 %	0.5 %	0.1 %
Anger	0.5 %	0.0 %	97.8 %	0.7 %	0.2 %	0.3 %	0.5 %
Disgust	0.0 %	0.0 %	1.0 %	98.4 %	0.0 %	0.3 %	0.3 %
Fear	0.2 %	0.6 %	0.4 %	0.4 %	96.7 %	0.2 %	0.5 %
Happiness	0.5 %	0.7 %	0.0 %	0.0 %	0.6 %	98.2 %	0.0 %
Sadness	0.6 %	0.3 %	0.2 %	0.0 %	0.5 %	0.1 %	98.3 %
Average				96.28 %			

Table 3: The confusion matrix with accuracy rates for each emotion category using the FER.

Results of the Fer2013 data set							
	Neutral	Surprise	Anger	Disgust	Fear	Happiness	Sadness
Neutral	74.1 %	2.2 %	1.2 %	1.3 %	8.6 %	3.5 %	8.3 %
Surprise	2.1 %	84.1 %	1.5 %	1.6 %	1.7 %	6.9 %	2.1
Anger	7.1 %	0.0 %	72.2 %	0.0 %	8.9 %	1.7 %	10.1 %
Disgust	4.0 %	2.1 %	9.1 %	73.2 %	5.4 %	3.9 %	2.3
Fear	2.4 %	12.2 %	1.0 %	3.2 %	71.1 %	0.0 %	10.1 %
Happiness	3.1 %	2.2 %	3.1 %	0.1 %	3.1 %	84.3 %	3.1
Sadness	9.1 %	1.7 %	8.2 %	0.0 %	8.7 %	1.5 %	70.8 %
Average				74.4 %			

Table 4: The confusion matrix with accuracy rates for each emotion category using the FER+.

Results of the FER+ data set								
	Neutral	Surprise	Anger	Disgust	Fear	Happy	Sad	contempt
Neutral	88.7%	2.1 %	0.0 %	0.1 %	3.5%	4.8 %	.08 %	0.0%
Surprise	6.9 %	87.1 %	0.0 %	1.8 %	0.7%	2.5 %	1.0 %	0.0%
Anger	6.9 %	0.0%	88.1%	0.0 %	0.0%	3.6 %	0.4 %	0.0%
Disgust	17.8%	3.2 %	21.8 %	55.9%	0.4%	1.0 %	0.0 %	0.0%
Fear	5.8 %	25.1 %	4.0 %	2.2 %	56.2%	0.0 %	6.7 %	0.0%
Happy	3.6 %	3.2 %	0.4 %	0.0 %	0.0%	92.8%	0.0 %	0.0%
Sad	15.7%	1.2 %	3.0 v	0.0 %	2.6%	0.8 %	73.7%	0.0%
	23.1%	1.8%	15.0 %	0.0 %	4.2 %	0.9 %	15.2%	39.8%
Average					85.1 %			

The second scheme enables training all weights of the large net and increases the accuracy rates by more than 4%. When training only the weights of the additional four layers results were 2% lower. It is important to notice, that when we used the second scheme for training, the accuracy rates of the model (D) ascended from 70.3% to 72.5% which did not happen with many rounds of direct training of this model. The same results with modest improvement (only 2.2%) were achieved on the FER+ data set with a final accuracy rate of 85.1%. As we also can see in Table 4, the proposed method performs well for the emotions Neutral, Happiness, Surprise, Sadness, and Anger, and worst for the remaining emotions. On the other hand, the data set have very few examples of these emotions and mostly with less quality. The total accuracy result are higher than the results achieved on the original *FER2013* data set due to the right re-tagging and the small size of the misclassified categories.

5. CONCLUSIONS

In this research, we present a novel architecture that parallelly aggregates three different ConvNets followed by additional four layers. This approach is a novel way of boosting existing architectures and by using the two-phases process of training, it enables better and faster conversion of the learning step. The two-phase training process increases the accuracy of the complete model by 3.5% on average and the accuracy of the basic models by 2% on average. In the scope of future work, we plan to target the same approach using a combination of a different number of the same basic models with different sizes and layer numbers. Using the same training process, we anticipate an improvement in the learning process in terms of speed and accuracy.

REFERENCES

- [1] D. Chan A. Mollahosseini and M. H. Mahoor. Going deeper in facial expression recognition using deep neural networks. The 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY., pages 1–10, 2016.
- [2] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd sourced label distribution. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, ICMI 2016, pages 279–283, New York, NY, USA, 2016. Association for Computing Machinery.
- [3] Peter Burkert, Felix Trier, Muhammad Zeeshan Afzal, Andreas Dengel, and Marcus Liwicki. Dexpression: Deep convolutional neural network for expression recognition, 2015.
- [4] Claude Chibelushi and Fabrice Bourel. Facial expression recognition: A brief tutorial overview 2002.
- [5] Rania Salah El-Sayed, Ahmed El Kholly, and Mohamed Youssri ElNahas. Robust facial expression recognition via sparse representation and multiple gabor filters. International Journal of Advanced Computer Science and Applications, 4(3), 2013.
- [6] FER2013. Challenges in representation learning: Facial expression recognition challenge facial-expression-recognition-challenge. <http://www.kaggle.com/c/challenges-inrepresentationlearning>, 2013.
- [7] FERPlus. Fer plus emotion label. <https://github.com/Microsoft/FERPlus>, 2016.
- [8] Yoav Freund and Robert E. Schapire. A decisiontheoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1): pages 119– 139, 1997.
- [9] Yijun Gan. Facial expression recognition using convolutional neural network. pages 1–5, 08 2018.
- [10] G.U.Kharat and S.V. Dudul. Emotion recognition from facial expression using neural networks. Human computer systems interaction advances in intelligent and soft computing, 2009.
- [11] Goodfellow I.J., Lee M., Hirose A., Hou Z.G., and Kil R.M. (eds). Challenges in representation learning: A report on three machine learning contests. Neural Information Processing, Lecture Notes in Computer Science, 8228, 2013.
- [12] Bo-Kyeong Kim, Jihyeon Roh, Suh-Yeon Dong, and Soo-Young Lee. Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. Journal on Multimodal User Interfaces., 10: pages 173–189, 2016.

- [13] S. Li and W. Deng. Deep facial expression recognition: A survey. *IEEE Transactions on Affective Computing*, pages 1–1, 2018.
- [14] Lin. Facial expression recognition based on geometric features and geodesic distance. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7(1): pages 323–330, 2014.
- [15] J. F. Cohn Lucey, J. Saragih T. Kanade, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotionspecified expression. In *CVPR Workshops*, pages 94–101, 2010.
- [16] M. Lyons, J. Budynek, and S. Akamatsu. Ieee trans. pattern analysis and automatic classification of single facial images. *Machine Intelligence*, 13: pages 252–263, 1991.
- [17] Alka Gupta M. and L. Garg. A human emotion recognition system using supervised self-organizing maps. In *IEEE International Conference on Computing for Sustainable Globle Development (INDIACOM)*, pages 654– 659, 2014.
- [18] Jharna Majumdar and Ramya Avabhrih. Human face expression recognition. *International Journal of Emerging Technology and Advanced Engineering Website*, 4(7): pages 559–565, 2014.
- [19] S. Mariooryad and C. Busso. Exploring cross-modality affective reactions for audiovisual emotion recognition. *Affective Computing, IEEE Transactions on*, 4(2): pages 183–196, April 2013.
- [20] Ekman P. and Friesen W. *Unmasking the face: A guide to recognizing emotions from facial expressions*. Consulting Psychologists press, palto Alta, CA, 1975.
- [21] Viola Paul and Jones Michael. Robust real-time face detection. *International Journal of Computer Vision*, 57(2): pages 137–154, 2004.
- [22] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: State of the art. *ArXiv*, 1612.02903, 2016.
- [23] R. Rosenthal. Conducting judgment studies: Some methodological issues. *The New Handbook of Methods in Nonverbal Behavior Research*, 2008.
- [24] R. Saabni. Facial expression recognition using multi radial bases function networks and 2-d gabor filters. In *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*., pages 225–230, 2015.
- [25] Akash Saravanan, Gurudutt Perichetla, and K. S. Gayathri. Facial emotion recognition using convolutional neural networks. *CoRR*, abs/1910.05602, 2019.
- [26] Yingli Tian, Takeo Kanade, and Jeffrey Cohn. *Facial Expression Recognition*, chapter 19, pages 487–519. 2011.
- [27] Matias Valdenegro-Toro, Octavio Arriaga, and Paul Pl'oger. Real-time convolutional neural networks for emotion and gender classification. In *27th European Symposium on Artificial Neural Networks, ESANN 2019, Bruges, Belgium*, 2019.
- [28] Z.-L. Ying, Z.-W. Wang, and M.-W. Huang. Facial expression recognition based on fusion of sparse representation. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 457–464, 2010.
- [29] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. In *ICMI 2015: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction.*, pages 435– 442, 2015.
- [30] S. Zafeiriou and M. Petrou. Nonlinear non-negative component analysis algorithms. *IEEE T-IP*, 19(4): pages 1050–1066, 2010.

AUTHORS

Raid Saabni is a senior lecturer and researcher at the school of computer science in the Academic College of Tel-Aviv Yafo and the TRD Center. He received his BSc in Mathematics and Computer Science in 1989 and his MSc and Ph.D. in computer science from the Ben-Gurion University of the Negev in 2006 and 2010, respectively. His research interests are: Machine learning, Computer vision, Historical Document Image Analysis, Handwriting Recognition, Image Retrieval, and Image and Signal Processing.



Alon Schcalar is a Senior Lecturer at the School of Computer Science at the Academic College of Tel-Aviv Yafo. He received his BSc, MSc (Summa Cum Laude) and PhD in Computer Science from Tel Aviv University. He co-authored over 30 papers in leading scientific journals and conferences. His areas of interest include machine learning (both unsupervised and supervised learning), dimensionality reduction, data mining, ensemble methods, image and signal processing and computer vision.



FOLLOW THEN FORAGE EXPLORATION: IMPROVING ASYNCHRONOUS ADVANTAGE ACTOR CRITIC

James B. Holliday and T.H. Ngan Le

Department of Computer Science & Computer Engineering,
University of Arkansas, Fayetteville, Arkansas, USA

ABSTRACT

Combining both value-iteration and policy-gradient, Asynchronous Advantage Actor Critic (A3C) by Google's DeepMind has successfully optimized deep neural network controllers on multi agents. In this work we propose a novel exploration strategy we call "Follow then Forage Exploration" (FFE) which aims to more effectively train A3C. Different from the original A3C where agents only use entropy as a means of improving exploration, our proposed FFE allows agents to break away from A3C's normal action selection which we call "following" and "forage" which means to explore randomly. The central idea supporting FFE is that forcing random exploration at the right time during a training episode can lead to improved training performance. To compare the performance of our proposed FFE, we used A3C implemented by OpenAI's Universe-Starter-Agent as baseline. The experimental results have shown that FFE is able to converge faster.

KEYWORDS

Reinforcement Learning, Multi Agents, Exploration, Asynchronous Advantage Actor Critic, Follow Then Forage

1. INTRODUCTION

In general, Machine Learning (ML) can be categorized into either supervised, unsupervised or reinforcement learning. Our work in this paper focuses on the last category. Stated simply, in reinforcement learning (RL) an agent gradually learns the best (or near-best) strategies based on trial and error which are performed through random interactions with the environment. The incorporation of the responses of these interactions help to improve the overall performance. Meaning the agents actions aim at both learning (explore) and optimizing (exploit). Exploitation is to make the best decision given current information whereas exploring is to gather more information. Many researches have been conducted to find the best strategies for the trade-off between exploitation and exploration. The trade-off between learning and optimizing is a classic problem in RL and is generally known as Exploitation versus Exploration.

There are many known methods for balancing between exploitation and exploration. When the state and action space is discrete, optimal solutions are possible. Bayesian RL [1] is an example of RL that can generate an optimal solution. However, when the state/action spaces are not discrete or the number of states grows very large, those previously optimal solutions become impractical. In these cases, we turn to heuristic approaches that are not perfect but are workable. The simplest approaches are random and greedy methods. With random choices the agent always chooses its action randomly during training. With greedy choices the agent always chooses its

action based on the best-known utility both during training as well as execution. The most commonly implemented non-ideal approach is the ϵ -greedy method. ϵ -Greedy exploration is a combination of random and greedy where the variable, ϵ , determines a rate at which the agent will choose randomly or choose greedily. Generally, as the agent learns the algorithm will decay ϵ towards zero, so that over time more exploiting and less exploring takes place. This ensures the agent can satisfactorily explore, while still acting nearly optimally when ϵ is very small.

In the area of RL the last few years have been filled with landmark achievements and ground breaking research [2]. In 2013, Mnih introduced Deep Q-Learning in the form of the Deep Q-Network (DQN) [3] [4]. Q-Learning [5] is an example of this type of RL. Q-Learning works by learning a utility function which we denote as $Q(s_i, \alpha)$ where the inputs of the function are states s_i and actions α . States here are the same as defined by Markov Decision Processes (MDP) [6]. This Q function defines the policy that will control the agent's actions. An optimal policy is a mapping from states to the corresponding actions the agent should take to maximum rewards in the long run. That is, it computes the action α for a given state s_i that will yield the highest discounted horizon utility. After learning has fully converged, the optimal policy will be known. The utility function in Q-Learning is updated with the following formula derived from the Bellman equation [7]:

$$Q(s_i, \alpha) \leftarrow (1-\eta^k)Q(s_i, \alpha) + \eta^k \left[r(s_i, \alpha, s_j) + \gamma \max_{\beta \in \mathcal{A}(j)} Q(s_j, \beta) \right] \quad (1)$$

In this formula, η is a learning rate, which controls how much the Q value is changed for each occurrence of a and i . The closer η is to 1, the faster the old Q value will be forgotten and replaced by the value computed by the rest of the formula. The term $r(s_i, \alpha, s_j)$ is the reward function, that for a given state s_i , action α and subsequent state s_j returns a reward value. The two states of the reward function are s_i and s_j where s_i is the current state where action α is performed to arrive in state s_j . The reward is added to the maximum Q value when we check each possible action β for state s_j and discount that Q value by the discount factor (γ). For γ , values near 0 prioritize immediate rewards, whereas values closer to 1 prioritize long-term utility. For most cases γ must be tuned to the problem at hand, but will always fall between 1 and 0. Over the course of training the factor k can be used to adjust the learning rate, which controls how much of the newer information derived from the sum of the reward and discounted future Q value replaces the old Q value. In fully deterministic environments a learning rate of 1 is optimal, where deterministic is defined as action α in state s_i always leads to state s_j . In stochastic environments the learning rate is decreased to zero over time, where stochastic is defined as there is a chance that action α in state s_i leads to state s_j . Because of this stochastic behavior it is not ideal to always replace the Q value with new information, so overtime as trust for the Q value grows it is changed less and less as the Q values are updated. Q values can be stored in a lookup table (Q -table). The Q -table can then be queried by the agent to make decisions based on the returned Q values. DQN built on top of Q-Learning by replacing the Q -table with a deep neural network (DNN).

DQN was able to reach human and beyond human level ability playing several specific Atari games. In 2016, Asynchronous RL [8] and specifically the development of A3C significantly improved previous efforts playing specific Atari games. A3C was able to produce better results than had been previously recorded by DQN and was able to learn much faster. A3C models its policy probabilistically where the policy output provides a probability for each possible action. These probabilities sum to 1 according to the current distribution of the model. In the case of A3C multiple agents choose actions nearly greedily (argmax of the policy output), but also seek to maximize an entropy term [9]. Entropy is used to skew the values used by the neural network

optimizer in a manner that encourages heterogeneity in the way it assigns probability to the possible actions. The purpose for entropy is to “encourage diversity” in the action selection, so that the algorithm doesn’t settle on a small select group of actions or action sequences. Entropy influences the error signal which influences how the model is trained. It causes the model to lean away from giving 100% of the probability to a specific action. This indirectly causes the agent to explore because it chooses actions probabilistically according to the distribution in its model.

Our research is an effort to improve A3C. Our contribution can be summarized as follows: While A3C demonstrates good training performance, we show that A3C’s training performance can be improved by adding FFE as an exploration strategy. FFE changes the way A3C chooses some of the actions to perform. In A3C the argmax of the policy output layer of its DNN is always selected, but as mentioned above that output is influenced by entropy so exploration is encouraged. FFE builds on top of that by controlling when to stop choosing the argmax of the output and choose an action randomly.

Analogous behavior can be observed in the physical world in ants. As ants search for food they begin by following a pheromone trail from ants that have gone before them, but as the pheromone grows weak, they start to explore on their own. Similarly, at the start of an episode of RL with FFE, an agent will exploit its knowledge to take actions that lead to higher rewards, but after following the reward trail for a variable length of time the agent stops exploiting and starts exploring. After some exploring following begins again. This leads to agents exploring more where exploration is most needed, which leads to faster learning times.

2. RELATED WORK

Exploration research in RL is not a new topic [10], but since the emergence of deep learning many new efforts have been made to improve this important aspect of RL. Because RL builds on the discovery of rewards, important to a models improvement. Many times rewards are very sparse or only take place at the end of the episode. In those cases generating intrinsic rewards can create stepping stones towards actual rewards. [11] [12] and [13] attempt to do this by adding additional DNNs to their systems structure. Those additional DNNs learn what part of the state space is well known and what part is unknown and generate an intrinsic reward to explore places that are less familiar to the model. Some of these researchers call this intrinsic reward, curiosity. Our method is much simpler and does not require the expensive cost of additional networks to improve learning.

Another classical system for improved exploration is keeping track of (or counting) every unique state the agent visits. In this way it can encourage the model to explore states that have a lower count or no count at all. The challenge here is that as states grow in size and complexity processing all this information is costly and the chances of the agent seeing all the possible states is unlikely. [14] and [15] try to overcome this challenge by using Monte Carlo tree search or hashing algorithms to estimate states. A historic achievement came as a result of related research that created AlphaGo [14], which was able to defeat some of the world’s best GO players. In the case of AlphaGo the number of possible states in a 19×19 GO board is: $\sim 2.082 \times 10^{170}$, but Monte Carlo searching combined with DQN proved capable of navigating this massive state space. Again, these efforts are considerably more costly and complex to implement than our simple method.

[16] described in detail the idea of Ant Colony Optimization (ACO) as a form of swarm intelligence. They explained how it could be applied to computer intelligence. Their applications for ACO are similar to FFE as they used it in a different domain.

Actor Critic RL [17] is similar to Q -Learning except with Actor Critic the policy and utility or value are separated into their own functions meaning the policy is independent of the value. In this case the policy is known as the actor and the value is known as the critic. The actor chooses actions and the critic critiques the actions. The critiquing is done by critic estimating a value at the start of an action or sequence of actions and comparing that with the actual value that was generated by the end of the action or sequence. The difference in the estimated value and the actual value is used as a signal that can be used to train both the actor and the critic. [18] showed how to generate a signal called an advantage A for state s and action a . Where Q is the same Q value as in Q -Learning and V is the value associated with given state s .

$$\text{Advantage} : A(s,a) = Q(s,a) - V(s) \quad (2)$$

[19] showed how to estimate the advantage instead of calculating Q values. The discounted reward that is used in calculating Q values is used as the replacement for the actual Q value. The formula for estimated advantage is the same as above except the Q function is replaced with the discounted reward.

3. ASYNCHRONOUS ADVANTAGE ACTOR CRITIC (A3C) - REVISITED

The three As of A3C stand for Asynchronous, Advantage, and Actor. The C of A3C stands for Critic. The algorithm is asynchronous because it relies on more than one agent playing the environment at the same time. For A3C, advantage, $A(s)$ is the estimated advantage [19]. The actor calculates the policy, $\pi(s)$, in the form of probabilities for each possible action for a given state in the form of a softmax output. The critic estimates the value of a given state $V(s)$ in the form of a linear output.

One of the benefits of A3C is that it uses many agents to explore different regions of isolated but equal environments, which is one of the reasons A3C learns much faster than its predecessor DQN. DQN [3] relied on a single agent while for their research, A3C, [8] utilized sixteen agents each running their own environment. A3C creates agents or workers that each have their own DNN and environment, such as an Atari game. Each agent operates in a separate thread, but they share the same actor and critic which are represented by a global DNN. However, the agents only operate on their own local copy of the global DNN. This network functions as both the actor and critic by using shared input and hidden layers but distinct output layers. One output layer is for the policy and the other output layer is for the value. At the start of a cycle each agent copies the global DNN over its local DNN, and collects experience as it plays the game. The experience is in the form of states, actions, and values. When the agents experience is large enough it is used to determine the discounted reward, R , and advantage, A . Once the discounted reward and advantage are known losses can be calculated for the value V and the policy π . The entropy H of the policy is also calculated.

$$\text{Value Loss} : L = \Sigma(R - V(s))^2 \quad (3)$$

$$\text{Policy Loss} : L = -\log(\pi(s)) * A(s) - \beta * H(\pi) \quad (4)$$

The entropy correlates with the spread of action probabilities output from the policy. When the probabilities are relatively equal entropy will be small, but when the probabilities are spread out the entropy is large. Entropy acts as a neutralizer that encourages the model to be conservative in regard to how strongly it thinks it knows the correct action. The agent takes the losses and uses them to calculate gradients that are used to optimize its local DNN parameters. The updated local DNN is then copied over the global DNN. This causes the global DNN to be constantly updated by the agents. This training process is repeated until convergence is detected. A3C relies solely on entropy to control exploration and exploitation. [8] has shown A3C is very capable, but our

modifications to A3C with FFE can improve its training performance. Adding FFE to A3C will force A3C to explore more by making more random choices and not only entropy-based choices.

4. OUR PROPOSED FOLLOW THEN FORAGE EXPLORATION

In our observation of RL learning, there are different times when more exploring or more exploiting offer more value to the learning process. For instance, when there isn't much progression made during a run, exploring at any time makes sense. We define the idea of a run as a training episode, and for our experiment a run is a single Atari game played from start to game over. If there isn't much progression the agent doesn't yet "understand" how different actions lead to different states and rewards. At this time maximum exploration will lead to the quickest increase in learning. If the agent has learned a meaningful path from start to end, then exploring at the beginning is less valuable than following the path and exploring later. When that path is sub-optimal, then additional exploration can introduce the agent to unfamiliar states. These experiences with unfamiliar states accelerate the agents learning. The central idea supporting FFE is that forcing exploration at the right time during a training episode can lead to improved training performance. Our implementation of FFE puts limits on exploiting and exploring. This is done to ensure the agent exploits closer to the beginning and end of a run, and is forced to explore in the middle.

Algorithm 1 describes our proposed algorithm which is an improvement version of A3C [8]. Our main contribution is shown at two steps, namely, Perform FFE and Update FFE. The first step of Perform FFE is detailed in the Algorithm 2 whereas Update FFE is described in Algorithm 3 as follows:

- **Algorithm 2 - Perform FFE.** This algorithm aims at selecting the best action α during training. FFE, when following, uses A3C's normal method of action selection. At a point in time when foraging should begin, FFE starts to select actions randomly. That point in time is determined by the Follow Action Limit m , and it is computed in Update FFE Algorithm 3. This forces FFE to perform more exploration than A3C normally would. However, too many random choices performed together in A3C will generally lead to poorer training performance, so the exploration must be limited. To enforce this limit FFE stops foraging and returns to following. This takes place after a controlled amount of exploration actions. This second Forage Action Limit n is also calculated in Update FFE (Algorithm 3). The injection of random exploration forces the agents into potentially less experienced states allowing the model to train faster. The other important aspect of this injection is that it happens approximately in the middle of the episode. It is our intuition that this will be the most valuable time in the episode for additional exploration to take place.
- **Algorithm 3 - Update FFE.** This step aims to keep track of a running value of the agents completed episode count o and average episode length p . For example, if FFE is used in a run this would be the number of actions required on average for the game to reach the game over state. At the end of each episode (i.e. game) p is calculated from previous experience and is multiplied by a random number percentage to set the Follow/Forage Action Limits, m and n .

After each action the agent will decrement one of its two Action Limits. If FFE is following it decrements its Follow Action Limit m value until that variable is zero. While m is greater than zero the agent always exploits. When m reaches zero the agent switches to Forage mode, where after each action the agent will decrement its Forage Action Limit n value until that variable is zero. While n is greater than zero the agent always explores randomly. Once n reaches zero the

agent returns to always following/exploiting. The use of the average episode length p ensures that the Action Limits m and n are dynamic and diverse in each episode.

[8] used multiple experiments to evaluate A3C, but the majority of those tests were playing various Atari games. That was accomplished through a program called the Atari Learning Environment (ALE) [20]. ALE is a simulator that can receive inputs that mimic Atari controller inputs and produce appropriate visual output that show an Atari game being played. ALE has many Atari games implemented. ALE itself has multiple implementations and for our research we used OpenAI's Gym (GYM) implementation of ALE. For our research, we used OpenAI's standard A3C implementation called Universe-Starter-Agent [21] as baseline during our benchmark and comparison against our proposed FFE.

In our experimentation we determined that best results were obtained with a large proportion of following and only a relatively small amount of foraging. The last if statement of the Algorithm 3 was added to ensure that foraging was controlled. We use the meta-parameters Φ and Ψ to limit the foraging. Parameter Φ controls the frequency of foraging actions. Then we use the difference of p and the value for m and scale that down by Ψ so that n is again limited. For our testing Φ was set to 0.5 and Ψ was set to 0.15. As stated, before this was to limit introducing too many poor action choices into the model. Also the 0.15 value for the scalar Ψ was used because m could be set to a potentially small value and then n could end up very large. This scalar ensured foraging never dominated following. The intuition for these limiting parameters was to allow the model to flex while foraging. By flex we mean that the model would not be presented with so many potentially poor action choices that the model was pulled away from improvement.

5. ALGORITHMS

Algorithm 1 Pseudo code of our improvement FFE to A3C. Our contribution is mainly shown at line 12, 18

```

1: //Assume global shared parameter vectors  $\theta$  and  $\theta_v$  and global shared counter  $T = 0$ 
2: //Assume thread-specific parameters vectors  $\theta'$  and  $\theta'_v$ 
3: //Assume thread-specific variables  $m$  (Follow Action Limit),  $n$  (Forage Action Limit),  $o$  (episode count)
    $p$  (average episode length)
4: Initialize  $m, n, o$  and  $p \leftarrow 0$ 
5: Initialize thread step counter  $t \leftarrow 1$ 
6: repeat
7:   Reset gradients:  $d\theta \leftarrow 0$  and  $d\theta_v \leftarrow 0$ .
8:   Synchronize thread-specific parameters  $\theta' = \theta$  and  $\theta'_v = \theta_v$ 
9:    $t_{start} = t$ 
10:  Get state  $s_t$ 
11:  repeat
12:    Perform FFE ▷ See Algorithm 2
13:    Receive reward  $r_t$  and new state  $s_{t+1}$ 
14:     $t \leftarrow t + 1$ 
15:     $T \leftarrow T + 1$ 
16:  until terminal  $s_t$  or  $t - t_{start} == t_{max}$ 
17:  if terminal  $s_t$  then
18:    Perform FFE Update ▷ See Algorithm 3
19:  end if
20:   $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \end{cases}$ 
21:  for  $i \in \{t-1, \dots, t_{start}\}$  do
22:     $R \leftarrow r_i + \gamma R$ 
23:    Accumulate gradients  $\theta'$  and  $\theta'_v$ 
24:  end for
25:  Perform asynchronous update of  $\theta'$  and  $\theta'_v$ 
26: until  $T > T_{max}$ 

```

Algorithm 2 Pseudo code of our proposed FFE to select best action α

```

1: //Assume thread-specific variables  $m$  (Follow Action Limit) and  $n$  (Forage Action Limit)
2: // $m$  and  $n$  were initialized to 0 in Algorithm 1
3: //Update FFE (Algorithm 3) will reset  $m$  and  $n$  to new values  $> 0$ 
4: if Follow:  $m > 0$  then
5:   //Follow Action Limit  $> 0$ : choose action normally
6:   Choose  $\alpha_t$  according to policy  $\pi(\alpha_t | s_t; \theta')$ 
7:    $m \leftarrow m - 1$ 
8: else
9:   if Forage:  $n > 0$  then
10:    //Forage Action Limit  $> 0$ : choose action randomly
11:    Choose  $\alpha_t$  from a uniform distribution in  $\alpha$ 
12:     $n \leftarrow n - 1$ 
13:   else
14:    //Both Action Limits = 0: return to choosing action normally
15:    Choose  $\alpha_t$  according to policy  $\pi(\alpha_t | s_t; \theta')$ 
16:   end if
17: end if
18: Perform  $\alpha_t$ 

```

Algorithm 3 Pseudo code of our proposed Update FFE

```

1: //Assume meta parameter  $\Phi$  (limits forage frequency) and  $\Psi$  (forage scaler)
2: //m and n = 0 when Update FFE is performed
3:  $U \leftarrow \text{Unif}$  ▷ Unif denotes the random number generator
4: Increment episode count:  $o \leftarrow o + 1$ 
5: Calculate average episode length:  $p \leftarrow (o - 1)/o * p + (1/o) * t$ 
6: Set Follow Action Limit:  $m \leftarrow U * p$ 
7: if  $U < \Phi$  then //Decide if Forage will happen in next training episode
8:   Choose not to Forage:  $n \leftarrow 0$ 
9: else
10:  Choose to Forage and set Forage Action Limit:  $n \leftarrow U * (p - m) * \Psi$ 
11: end if

```

6. EXPERIMENTS

To validate our research we ran numerous tests comparing a default implementation of A3C with a version of A3C modified with FFE. For our benchmark we used OpenAI's Universe-Starter-Agent [21], and for our modified version we added FFE to the default [21] implementation. The environments tested were various Atari games as implemented by OpenAI's GYM.

All our experiments were conducted on a Microsoft Azure Data Science Virtual Machine for Linux. Some initial tests were conducted using 8 cpu virtual machines, but for all the recorded experiments in this paper a Standard DS5 v2 Promo (16 vcpus, 56 GB memory) virtual machine was used. A3C was configured with all the default settings from [21]. For each experiment three test runs were done, and the results were averaged together for the results presented. For the Atari game, Pong, the score is calculated based on the total score of the agent minus the total score of the computer opponent. The game/episode is over when either the agent or computer opponent achieves a score of 21. Figure 1 shows A3C modified with FFE (indicated in blue) outperforms the baseline A3c (orange) by reaching a higher score faster. For this test entropy was left at the default value. This result shows that FFE was able to speed up the learning process for A3C. FFE created a much steeper and stable rate of learning. This shows that the intuition for FFE has merit and more extensive testing is warranted.

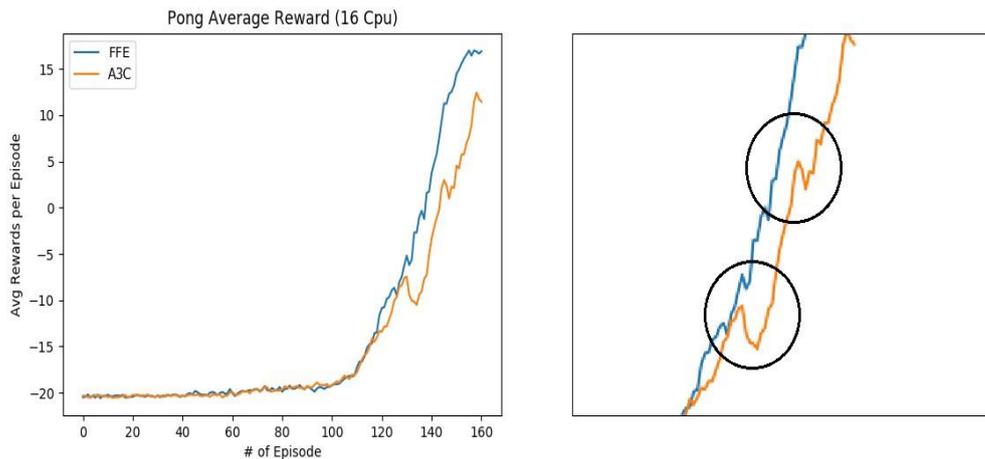


Figure 1: shows the average reward all the agents achieved per episode playing Atari Pong. The right side shows a section of the same chart highlighting the improved stability of FFE. This chart limits the results to each algorithm performing 4 million global steps.

We also tested a range of different entropy scalars to see how entropy affected the performance of the algorithm. Figure 2 shows that when entropy is not used FFE alone is not sufficient to generate a good result (labeled as 0.0). Figure 2 also shows when we tried scaling entropy by .001 (labeled as 0.001) instead of the default .01 (labeled as 0.01). We found that both .001 and .01 learned in about the same number of episodes total, but by scaling entropy by .001 this caused the agents to play less optimally and thus the agents took more actions to complete the episodes, which can also be seen in Figure 2. The curves of the average game length rise and then fall because the game takes longer when both players scores get close to 21, but then as the agent starts to play much better than the computer opponent the average length starts to decrease until it plateaus when the agent is winning episodes 21 to 0. This experiment shows that FFE alone is not able to replace entropy as an exploration strategy.

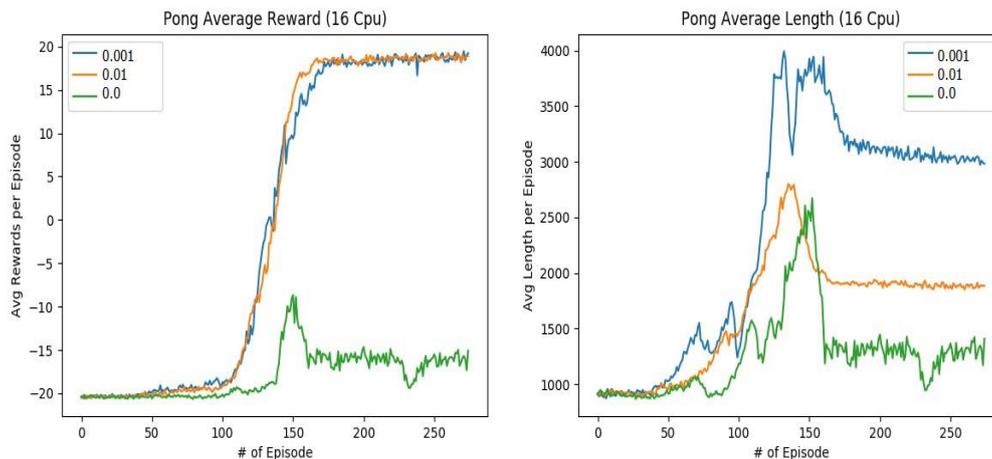


Figure 2: shows how entropy scaling effects training performance for AC3 modified with FFE. The entropy value is scaled according to the key values. [8] used a value 0.01 as the scalar value. This chart limits the results to each algorithm performing 10 million global steps.

Our next experiments involved more challenging Atari games. We tested Boxing, Amidar, and Beamrider. Figure 4 shows the results of experiment of the baseline A3C and A3C modified with FFE playing Atari Boxing. Boxing has a maximum score of 100. When either player punches the other player successfully they are rewarded with a point, and the game is over when either opponent reaches a score of 100 or time runs out. The final score is the agent's score minus the computer opponents score. While the results of both algorithms are close, FFE narrowly outperforms baseline A3C by reaching a higher score faster. Figure 5 shows the results of the experiment with the same algorithms playing Atari Amidar. Amidar has no maximum score. That being the case our agents did not score very high. Still our results were better than the high score achieved by A3C LSTM [8] which was 176 after four days of training. Lastly, Figure 6 shows the results of the experiment with the same algorithms playing Atari Beamrider. In this experiment FFE failed to outperform baseline A3C. This result is perhaps due to instability in the models DNN. Figures 3, 4 and 5 are found in the appendix. In our tests training 100 million global steps of A3C took approximately 24 hours. The cost associated with that much Microsoft Azure virtual server usage limited how much we could train.

The documentation concerning our baseline A3C states that the algorithm is tuned for good Pong performance. Meaning baseline A3C might perform poorly on other Atari games. Here is a list of some differences between [8] and [21] implementation of A3C. The original A3C used a shared optimizer for all agents, and baseline A3C uses distinct optimizers for each agent. Also to note

baseline A3C is designed to be able to play games in real time, so it stores experience in a separate process while the optimizer ran, and the original A3C would force the agents to wait while the optimizer ran.

7. CONCLUSIONS

FFE demonstrates that relying on entropy alone is not the most efficient method to train A3C. Exploration utilizing FFE can allow the learning process (training) to be improved. We compared a version of A3C equipped with FFE against the default A3C with several different Atari games, and found that FFE improved results in the majority of cases. These results are promising, and provide evidence that FFE improves the default exploration strategy utilized by A3C.

When training Pong we were able to reach the maximum score in less than forty minutes for a best case. [8] listed a Pong training time at taking two hours. [8] showed the result of a good score for Breakout in less than four hours, in our research we did not have meaningful results after twenty four hours of training on Breakout. For future work we plan to utilize a more cost-efficient computing environment to allow training on more diverse and difficult environments.

For future work we will research moving FFE's forage to other times during the episodes. Perhaps foraging at the beginning or end of the episode could lead to better performance. We will evaluate an implementation of A3C that uses a shared optimizer instead of distinct optimizers.

Entropy plays an interesting role in all the experiments. For future work we plan to analyze entropy more closely. FFE also utilizes meta-parameters. We set Forage to only happen in half of the episodes and scaled the computed Forage value smaller. Because of these meta-parameters more testing is needed to determine their optimal values.

REFERENCES

- [1] M. Ghavamzadeh, S. Mannor, J. Pineau, A. Tamar and others, "Bayesian reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 8, p. 359, 2015.
- [2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, p. 85, 2015.
- [3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing atari with deep reinforcement learning," in *NIPS Deep Learning Workshop*, 2013.
- [4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski and others, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, p. 529, 2 2015.
- [5] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, p. 279, 1992.
- [6] R. Bellman, "A Markovian decision process," *Journal of Mathematics and Mechanics*, p. 679, 1957.
- [7] R. Bellman, *Dynamic programming*, Courier Corporation, 2013.
- [8] V. Mnih, A. Puigdomènech, M. M. B. Badia, A. Graves, T. P. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *33rd International Conference on Machine Learning (ICML)*, 2016.
- [9] R. J. Williams and J. Peng, "Function optimization using connectionist reinforcement learning algorithms," *Connection Science*, vol. 3, p. 241, 1991.
- [10] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, p. 237, 1996.
- [11] B. C. Stadie, S. Levine and P. Abbeel, "Incentivizing exploration in reinforcement learning with deep predictive models," *arXiv preprint arXiv:1507.00814*, 2015.
- [12] R. Houthoofd, X. Chen, Y. Duan, J. Schulman, F. De Turck and P. Abbeel, "Vime: Variational information maximizing exploration," in *Advances in Neural Information Processing Systems*, 2016.

- [13] D. Pathak, P. Agrawal, A. A. Efros and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” in *International Conference on Machine Learning (ICML)*, 2017.
- [14] D. Silver, A. Huang, A. G. Chris J. Maddison, L. Sifre, G. V. D. Driessche, J. Schrittwieser and others, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, p. 484, 1 2016.
- [15] H. Tang, R. Houthoof, D. Foote, A. Stooke, O. X. Chen, Y. Duan, J. Schulman, F. DeTurck and P. Abbeel, “# Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, 2017.
- [16] M. Dorigo, M. Birattari and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, p. 28, 11 2006.
- [17] I. Grondman, L. Busoniu, G. A. D. Lopes and R. Babuska, “A survey of actor-critic reinforcement learning: Standard and natural policy gradients,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, p. 1291, 2012.
- [18] L. C. Baird III, “Advantage updating,” 1993.
- [19] R. S. Sutton, D. A. McAllester, S. P. Singh and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Advances in neural information processing systems*, 2000.
- [20] M. G. Bellemare, Y. Naddaf, J. Veness and M. Bowling, “The Arcade Learning Environment: An Evaluation Platform for General Agents,” *CoRR*, vol. abs/1207.4708, 2012.
- [21] G. Brockman, C. Olsson and A. Ray, *Universe Starter Agent*, 2016.

APPENDIX

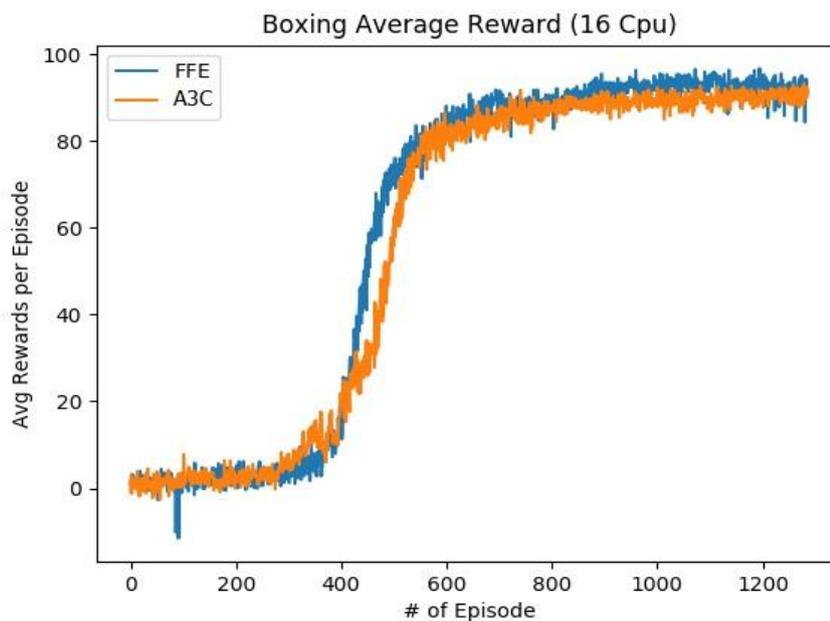


Figure 3: shows the average reward all the agents achieved per episode playing Atari Boxing. This chart is limited to reporting the results of each algorithm performing 40 million global steps.

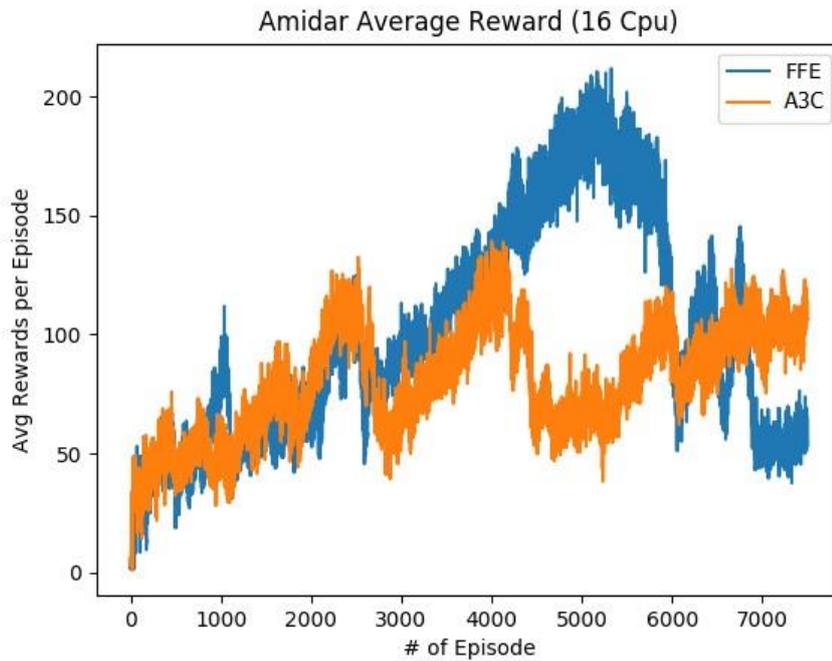


Figure 4: shows the average reward all the agents achieved per episode playing Atari Amidar. This chart is limited to reporting the results of each algorithm performing 60 million global steps.

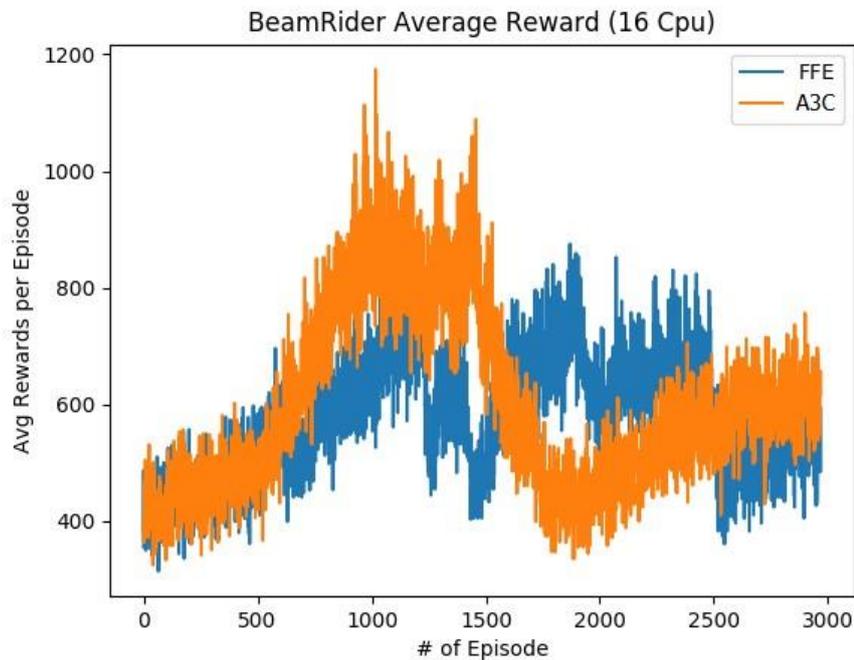


Figure 5: shows the average reward all the agents achieved per episode playing Atari Beamrider. This chart is limited to reporting the results of each algorithm performing 100 million global steps.

AUTOMATED CLASSIFICATION OF BANANA LEAF DISEASES USING AN OPTIMIZED CAPSULE NETWORK MODEL

Bolanle F. Oladejo and Oladejo Olajide Ademola

Department of Computer Science, University of Ibadan, Ibadan, Nigeria

ABSTRACT

Plant disease detection and classification have undergone successful researches using Convolutional Neural Network (CNN); however, due to the intrinsic inability of max pooling layer in CNN, it fails to capture the pose, view and orientation of images. It also requires large training data and fails to learn the spatial relationship of the features in an object. Thus, Capsule Network (CapsNet) is a novel deep learning model proposed to overcome the shortcomings of CNN. We developed an optimized Capsule Network model for classification problem using banana leaf diseases as a case study. The two dataset classes include Bacterial Wilt and Black Sigatoka, with healthy leaves. The developed model adequately classified the banana bacterial wilt, black sigatoka and healthy leaves with a test accuracy of 95%. Its outperformed three variants of CNN architectures implemented (a trained CNN model from scratch, LeNet5 and ResNet50) with respect to rotation invariance.

KEYWORDS

Capsule Network, CNN, Activation function, Deep Learning, Precision Agriculture.

1. INTRODUCTION

In the recent years, there has been a significant upsurge in the use of computer vision techniques for image processing in precision agriculture. Banana (*Musa Acuminata*) is one of the most consumed staple food in Nigeria. It has great impact in food security and provide high intake of carbohydrates and calories for more than 70 million people in Sub-Saharan Africa [9]. However, there are some diseases affecting banana produce in Sub-Saharan Africa such as Banana Black Sigatoka, Banana Bacterial Wilt and Banana Streak etc. Farmers spend a lot of money to hire plant pathologist who manually checked the banana leaf in order to spot the diseases and provide a measure to control them. This process is prone to error and bias, hence the need for an automatic detection of these diseases using Artificial Intelligence techniques. Deep learning is a domain in Artificial Intelligence which automatically explore the features of data to perform a given tasks such as, computer vision, natural language processing etc.. It is a subject that has been explored in early days to solve vision problems such as object detection, image classification, disease diagnosis and classification etc. A state-of-the-art performance had been achieved using various deep learning techniques on a benchmark dataset such as ImageNet [8]. Convolutional Neural Network (CNN) is one of the various artificial intelligence techniques which has been explored extensively in precision agriculture. However, given the unavailability of plant pathologist or agronomist in rural area, it is very challenging to get large amount of annotated diseased plant and healthy leaves dataset. The limited dataset issue causes CNN models to overfit which was why data augmentation techniques were employed to increase the size of dataset.

The operations of CNN architecture does not consider the spatial relationship between the lower and higher level features of an object.. That is, they tend to lose information about feature locations, the spatial relationship of features and orientations of features in the input data. Hence, there is a need to design a deep learning model which could perform classification problem with better accuracy on small dataset. CNN architectures also fail to model the equivariance of an object and this is why a novel deep learning model was introduced by [18], referred to as Capsule Network (CapsNet) which can be seen as a group of neurons which models the hierarchical relationships of features or objects. This object have different properties which includes the pose (position, size, orientation), deformation, velocity, albedo, hue, texture, among others.

CapsNet accepts input as a vector and it consists of Convolutional layer, Primary Capsule layer, DigitCaps layer, and the encoder-decoder. A capsule is designed as an activity vector of an input image whose overall length represents the probability or likelihood of the existence of an entity and its orientation represents its instantiation parameters. The capsule comprises a bunch of neurons stacked in capsules. The primary idea of Capsule Network is to replace the pooling algorithm of CNN with a Dynamic Routing Algorithm. Capsule Network is suitable for classification problem because of its ability to model the 3D structure of an object with the concept of inverse graphics [6]. CapsNet has achieved state-of-the-art performance on MNIST Handwritten Digit Recognition dataset by reducing the test error benchmark of CNN. However, with the current performance of capsule network, there is a room for improvement in order to augment its distinction on complex images (Three colour channel images). To the best of our knowledge, this is the first paper applying Capsule Network in Precision Agriculture. In order to augment the distinction of Capsule Network in terms of accuracy, ensuring faster convergence and reducing the training time, in this paper, we propose an optimized version of CapsNet for Banana Leaf Disease Classification model.

The organization of this paper is as follows. Section (2) discusses the related works, and in section (3), we discuss the methodology of the optimized capsule network for banana leaf disease classification. Section (4) highlights the results and evaluation metric used for the designed model and the paper is concluded in Section (5).

2. RELATED WORKS

In this section, we review the related works in classification tasks using deep learning architectures. Generally, researchers have extensively explored the use of deep learning techniques for object recognition and image classification tasks [3, 4, 13, 15]. Convolutional Neural Networks is a deep learning algorithm that has been used in recognition and classification problem and has achieved state of the art performances in image classification [10]. [17] developed the first CNN architecture named LeNet-5 for object detection and was evaluated with the handwritten digit MNIST dataset. The LeNet-5 architecture has 7 layers, which all contain trainable parameters (weights). It has two sets of convolutional, average pooling layers, flattening convolutional layer, two fully connected layers and the softmax classifier. It was designed before the advent of Graphics Processing Units (GPU) and one of the bottlenecks of this architecture is that it takes a longer time to train. In [2], an improved LeNet-5 architecture to classify two classes of banana leaf diseases was proposed. The limitations of the LeNet-5 architecture was opined by [14] and a deeper and wider version of the LeNet which was proposed won the ImageNet ILSVRC-2012 challenge with a test error rate of 15.3%. AlexNet has been applied to various image classification task [11, 19]. Other CNN architectures are also developed in order to reduce time complexity and number of trainable parameters.

It was noted in [3] that, the factors influencing the use of deep learning algorithms for plant disease detection and classification includes insufficient size and variety of annotated dataset which

affect the performance and the generalizability of the deep learning models. However, a better way to overcome this shortcoming is to use transfer learning on a pre-trained network or perform data augmentation using Generative Adversarial Networks (GANs) [5, 13]. Another point raised by [3], is the symptom representation on plant leaves, Image background, multiplesimultaneous disorder, symptoms variation, disorders with similar symptoms and the covariate shift which is the phenomenon in which differences between the data used in training the model and the data on which the model is tested on results to low accuracies. The covariate shift problem was observed by [18] where their model which has an accuracy of 99.5% drops drastically to 31% because the model was tested with a dataset from the cultivated field after being trained with a laboratory data.

In general, CNNs algorithms have been extensively explored in computer vision tasks but there are some shortcomings of CNNs that needs to be addressed. [6] gave some arguments about the limitations of convolutional neural networks which are: (i) CNN architectures are weak to affine transformation and do not consider much about the spatial relations of the features in the image,

(ii) The max-pooling layer in CNN tends to lose information while routing features from one layer to another, (iii) They fail to model the rotation invariance of an object. [18] proposed a Capsule Network with Dynamic Routing algorithm to overcome the drawbacks of CNN architecture. In [21], the capsule network architecture for gait recognition in which two operations were performed namely, matching local features at the bottom layer and matching mid-level features at the middle layer based on capsule network, was used, with the CASIA-B dataset and OUISIR Treadmill dataset B. Some researchers have also used capsule network on medical imaging for classification of diseases [1] and they have achieved better accuracies compared to the conventional CNN.

These above mention methods all utilized Image processing techniques to manipulate images and extract features for detection and classification task. However, the challenge with these methods is their inability to perform well on unconstrained imaging conditions like data gotten from the cultivated field. The dynamic routing by agreement of the capsule network makes it slower to train and, in a bid, to solve this challenge, [8] proposed a Matrix Capsule with Expectation-Maximization (EM) algorithm which uses a pose matrices. The routing process is similar to fitting a mixture of Gaussians using Expectation-Maximization, where each capsule in the in the lower layer is routed to a capsule with a similar vote in the higher layer. A novel deep learning approach for traffic sign classification using capsule network was also developed by [16]. Their model was trained and tested on German Traffic Sign Recognition Benchmark dataset (GTSRB) and they achieved a state-of-the-art accuracy of 97%. Capsule networks have also been explored in medical imaging task and have achieved good accuracy compared to conventional convolutional neural network techniques. [1] used capsule network to classify brain tumor type on MRI images. They investigate the overfitting problem and also design a visualization paradigm to view the features learned from the MRI image dataset. Motivated by the performance of capsule network, we are proposing an optimized version of capsule network for banana leaf disease classification by improving the network in order to reduce the training time without a tradeoff for the accuracy of the network. To the best of our knowledge, this is the first work done in the problem domain.

Table 1. Comparative literature review of existing model/framework for image processing using deep learning algorithm.

Paper	Description	Algorithm/Method
[2]	A deep learning based approach for banana leaf disease classification. They Classify two (2) fungi banana leaf diseases (Black Sigatoka and Banana Speckle) with CNN based on LeNet5 architecture.	LeNet-5 a variant of CNN architecture was used to detect two types of banana leave disease.
[17]	Using Deep Learning for Image-Based Plant Disease Detection. They trained a convolutional neural network with 54,306 images of diseased and healthy plant leaves collected under controlled conditions to identify 14 crop species and 26 diseases.	A CNN model was trained to identify 14 crop species and 26 diseases.
[15]	Capsule Networks for Traffic Sign Classification was developed. They apply capsule network to traffic sign classification on German Traffic Sign Recognition Benchmark dataset (GTSRB). They achieved better performance and accuracy on the training and testing data compared to the conventional CNN architectures.	A Capsule Network model was developed to detect traffic signs.
[22]	A novel three-channel convolutional neural networks (TCCNN) model was developed by combining three color components of RGB for vegetable leaf disease recognition.	CNN model was developed which makes used of the softmax layer for feature vector to classify the input veget- able leaf images.
[1]	Brain tumour type classification using capsule network. They design a capsule network algorithm to investigate the overfitting problem with CNN algorithms on MRI images.	Capsule Network for brain tumour classification

3. MATERIALS AND METHOD

The focus of this work is to design an Optimized Capsule Network model for banana leaf disease classification which will aid a faster detection and classification of banana leaf diseases. The following sections analyses the technique and approach used in implementing the Optimized Capsule Network model.

3.1. Dataset

The database was formed by collecting Banana Leaf disease and healthy images namely: Banana Bacterial Wilt disease (*Xanthomonas wilt*), Banana Black Sigatoka disease (*Mycosphaerella fijiensis*), and Healthy banana leaves. Three class labels were assigned to the collected dataset. We resized all the images collected to 256 x 256 pixels, preprocess and perform the training, validation and testing on the images. In all our experiments, we use the dataset with three colour channels (Red, Green, and Blue) without converting them to grayscale. 1000 images were collected on the cultivated field across the three classes and the collected images were divided in the ratio of 80%:20% for training and testing respectively. The 20% for testing was

further divided in the ratio of 90%:10% for testing and validation respectively. Below is the sample image of the three classes of the dataset:

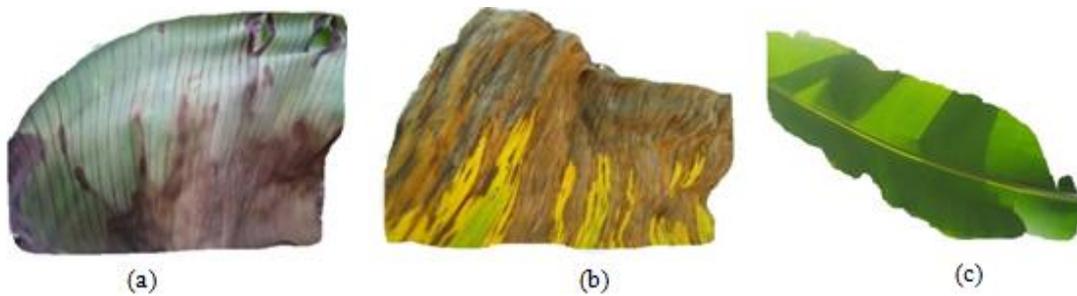


Figure 1. Sample of the banana leaf images collected. a. Banana Bacterial Wilt Disease image, b. Banana Black Sigatoka Disease image and c. Healthy leaf image.

3.2. Data Preprocessing

The dataset was collected and annotated with the aid of a plant pathologist. The annotated datasets were fed directly to the model without manual feature engineering to extract features. The data collected has no uniformity and in order to solve this problem, we applied data augmentation technique to the dataset in order to have a balanced data. Figure 2, Shows the flow of datasets collection, annotation and augmentation.

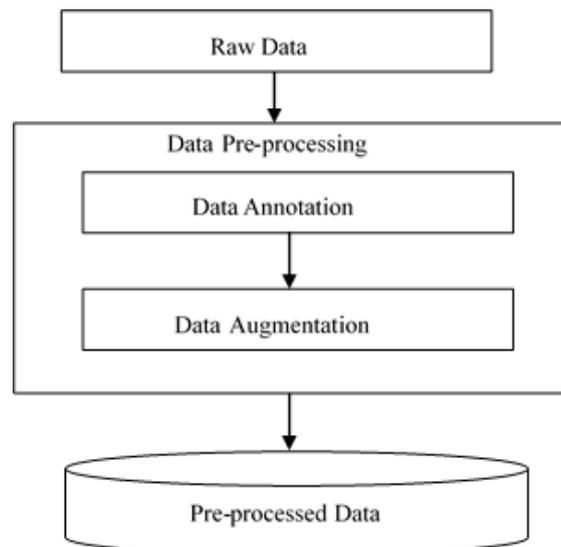


Figure 2. Schematic Diagram of Data Preprocessing.

3.3. Development of an Optimized Capsule Network Model

The Optimized Capsule Network (OCapsNet) model consists of the Input layer, Convolution layer, Primary capsule layer, DiseaseCaps layer and the fully connected layer. Figure 3 below shows OCapsNet architecture. The raw data of the input layer are down-sampled from 256 x 256 to 128 x 128 in order to reduce the number of parameters and in turn reduce the training time.

The first layer of our model is the input which consists of a three colour (Red, Green, and Blue) channel images of banana bacterial wilt disease, banana black Sigatoka leaf disease and healthy

leaves with an input size of $128 \times 128 \times 3$. The second layer is the convolutional layer with Leaky-ReLU activation function to extract the features from the input data. In the convolutional layers, multiple kernels are used to extract the features from the banana leaf images and every kernel produces a feature map. In order to reduce dimensionality, we use convolution with a stride of 2 (The dimension is reduced by the factor of 2). The first convolutional layer has 256 channels, kernel size of 9×9 , stride of 1, and no padding while the second convolutional layer has 256 layers, kernel size of 9×9 , stride of 2, and no padding. The third layer is the Primary Capsule (Primary Caps) layer in which the output of the convolutional layer is fed into the layer which contains multiple neurons at the same location in different feature maps grouped in capsules. The capsules in the primary capsule layer are connected with all the capsules in the Disease Caps layer where a dynamic routing-by-agreement algorithm is applied to check if the low-er level features learned is in agreement with the higher-level features. The dynamic routing-by agreement algorithm in figure 4 is a better alternative to the max-pooling operation in conventional Neural Networks (CNNs) as it tends to lose information while routing features from one layer to the other. The Disease Caps layer consists of three classes and the output is used to make a decision about the class of the input image. The number of neurons was reduced in the fully connected layer and an optimizer with momentum is used in order to have a faster network. The decoder consists of two fully connected layers with 512 and 1024 neurons respectively. We re-duced the number of the fully connected layer in the default architecture from three to two in other to reduce the training time of the network.

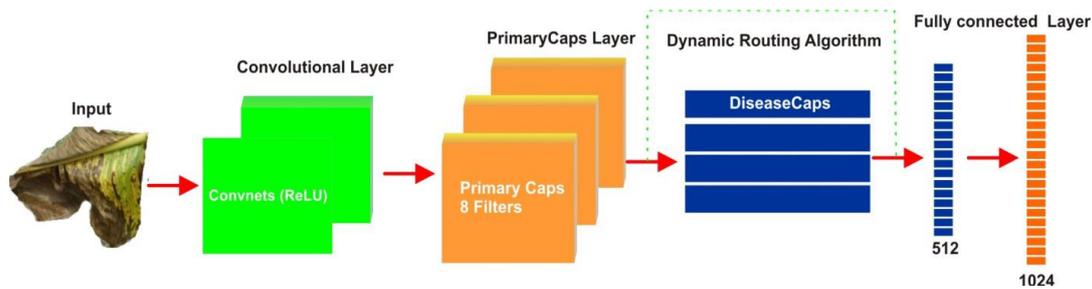


Figure 3. A schematic diagram of optimized capsule network model.

3.3.1. Dynamic Routing Algorithm

In capsule networks, the first convolution layer learns the basic features of an object and there is a need to route the features learned from the lower-level layer to the appropriate parent capsule representing the path-whole in the higher-level layers. The effective transfer of the features learned from one layer to another is made possible by the dynamic routing algorithm, the routing-by-agreement, which decides which entity from the lower level capsule, agrees with the output of the higher-level capsule through an iterative process. This process is done by a prediction or vote of a capsule i in the lower layer i for a capsule j in higher layer which is produced by multiplying its pose by a weight matrix W_{ij} .

$$\tilde{U}_{jfi} = W_{ij}u_i \quad (1)$$

Where \tilde{U}_{jfi} is a prediction vector gotten from the multiplication of the lower level vector U_i and the weight matrix W_{ij} . The prediction is computed through a transformation matrix, and when multiple predictions from low-level capsules agree with the output of the high-level capsule, this gets activated. In order to get the weighted sum S_j , the prediction vector is multiplied to the coupling coefficient C_{ij} which is determined by the routing-by-agreement algorithm. The routing algorithm calculates the agreement between the lower level vector and the higher level vector by doing a scalar product (\cdot) of the prediction vector and the coupling coefficient. If the prediction

vectors agree with each other, then the coupling coefficient will be higher which means, the two capsules are relevant to each other.

$$S_j = \sum_i C_{ij} \tilde{U}_{ji} \quad (2)$$

Where C_{ij} is coupling coefficients that are calculated dynamically by the iterations of the dynamic routing. The coupling coefficients between one capsule and the others are sum to one and determined by a routing Soft Max whose initial log its b_{ij} is the log prior probabilities that capsule in one layer should be coupled with the capsule in another layer. The agreement between these capsules is simply the scalar product of $a_{ij} = v_j \cdot \tilde{U}_{ji}$.

$$C_{ij} = \text{softmax}(b_{ij}) = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (3)$$

The weighted sum S_j is passed through a nonlinearity with a squash function in order to ensure that the vector length is between 0 and 1. That is, if the length of the vector is 1, there is a probability that a feature exists in the image.

$$V_j = \frac{\frac{|S_j|^2}{1 + |S_j|^2} S_j}{|S_j|} \quad (4)$$

Algorithm 1. Dynamic Routing Algorithm procedure as described by [21].

-
1. **Procedure** ROUTING (\tilde{U}_{ji}, r, l)
 2. **For** all capsule i in layer l and capsule j in layer $(l + 1)$: $b_{ij} \leftarrow 0$.
 3. **For** r iterations **do**
 4. for all capsule i in layer l : $c_i \leftarrow \text{softmax}(b_i)$
 5. for all capsule j in layer $(l + 1)$: $s_j \leftarrow \sum_i C_{ij} \tilde{U}_{ji}(b_i)$
 6. for all capsule j in layer $(l + 1)$: $v_j \leftarrow \text{squash}(s_j)$
 7. for all capsule i in layer l and capsule j in layer $(l + 1)$: $b_{ij} \leftarrow b_{ij} + v_j \cdot \tilde{U}_{ji}$
 8. **return** v_j
-

3.3.2. Loss Function

In order to train the network, two losses are employed which are, the margin loss and the reconstruction loss. The basic idea of the margin loss is to compute the error signal based on the length of the output vector of the DiseaseCaps. Hence, for the network to be able to segment highly overlapping objects, a loss is introduced in order to allow multiple object instances of different classes.

$$L_k = T_k \max(0, m^+ - \left\| \left| v_k \right| \right\|^2) + \lambda (1 - T_k) \max(0, \left\| \left| v_k \right| \right\| - m^-)^2 \quad (5)$$

Where $T_k = 1$ if and only if digit k is in the image, $m^+ = 0.9$, $m^- = 0.1$ and $\lambda = 0.5$ is a down-weighting parameter for absent digit classes that stops the initial learning from shrinking lengths of the activity vectors of all digit capsules. v_k is the capsule that represents digit k if a digit exists,

v_k will be high and the loss will be lower and if v_k is low, there would be a high loss. The total loss used for training is the sum of all margin losses for each digit defined by $\text{Loss} = \sum_k L_k + 0.0005R$.

4. EXPERIMENT AND RESULT

The model was trained, validated, and tested with the preprocessed dataset on a Google AI Cloud Service (Google Collaboratory) with the following configurations: Graphics Processing Unit (GPU) 1xTesla K80, having 2496 CUDA cores, computer 3.7, 12GB (11.439GB Usable) GDDR5 VRAM, CPU 1x single core hyper-threaded i.e. (1 core, 2 threads) Xeon Processors @2.3Ghz (No Turbo Boost), 45MB Cache, RAM 12.6 GB Available, Disk 320 GB Available. The model was implemented using Python programming language, TensorFlow as the backend and Keras library in Python 3.7. We ran the capsule network training 7 times by creating seven models, while we achieve the best result on the seventh model with an input size of 128 x 128.

4.1. Evaluation Metric

Accuracy evaluation metric was used in order to evaluate the proposed model. It is the proportion of the total number of predictions that were correctly classified i.e. the ratio of correctly predicted observation to the total observation.

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} \quad (6)$$

Where True Positive rate (TP) and True Negative rate (TN) are correctly classified while False Positive (FP) and False Negative (FN) are correctly misclassified.

4.2. Experimental Result

The proposed capsule network model achieved a test accuracy of 0.953642 (95%) using the collected datasets. Figure 4 and 5 below show the accuracy and loss graph of the proposed model. We also implemented a CNN model from scratch, LeNet5 and ResNet50 architecture and their results are reported in Table 2.

Table 2. Test Accuracy of Capsule Network, CNN, LeNet5 and Resnet50 architecture.

Architectures				
Models	Our model	CNN	LeNet5	ResNet50
Accuracy	95%	89%	82.5%	96%

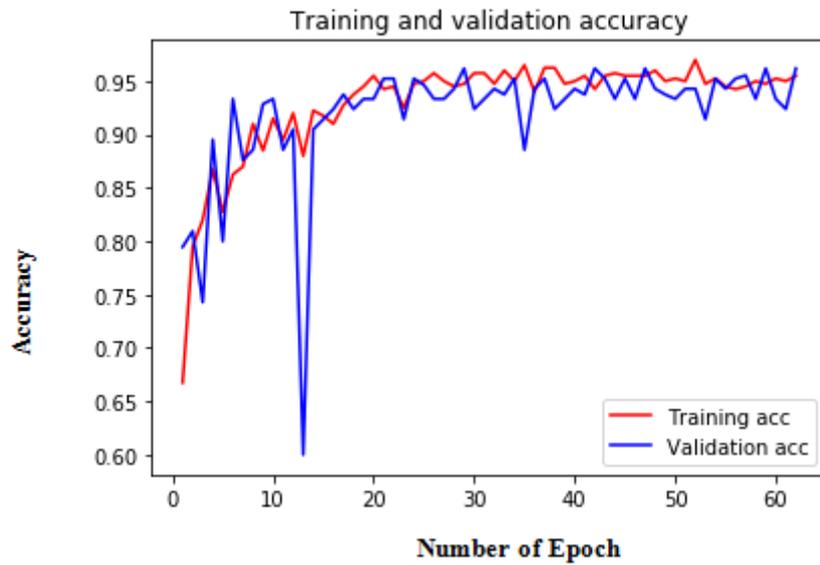


Figure 4. A graph showing the accuracy and validation of the optimized capsule network

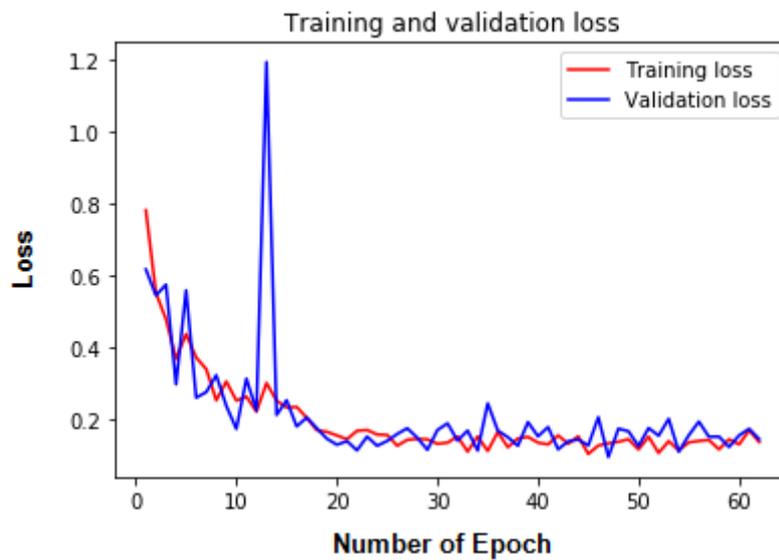


Figure 5. A graph showing the training and validation loss of the optimized network.

In order to compare our model with existing architectures, we implemented a CNN architecture from scratch, LeNet5 architecture and ResNet50 architecture using transfer learning. The CNN architecture achieved a test accuracy of 89%, LeNet5 architecture achieved a test accuracy of 82.5% while the ResNet50 architecture using transfer learning achieved a test accuracy of 96%. Table 2 above shows the results of the implemented architectures. We further rotate the test dataset in 30, 60, and 90 degrees and used it to test the implemented models. Table 3 below shows the result obtained after the training dataset were rotated. Figure 6 further illustrates the visualization of the result of comparison among the different models.

Table 3. Results of the Optimized Capsule Network, CNN, LeNet5 and ResNet 50 architecture when the test datasets were rotated to 30, 60 and 90 degrees.

Model	Dataset without rotation	Test dataset with 30 degrees rotation	Test dataset with 60 degrees rotation	Test dataset with 90 degrees rotation
Capsule Network	95%	93%	95%	95%
CNN	89%	67.5%	59%	85.5%
LeNet5	82.5%	77%	78%	77%
ResNet50	96%	75%	76%	72%

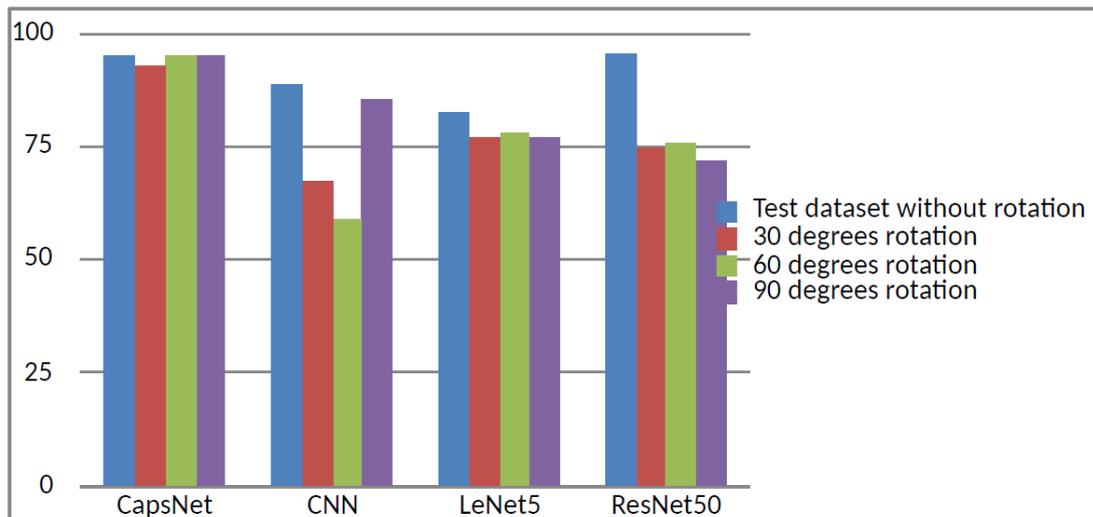


Figure 6. A bar chart showing the comparison of the implemented models.

5. CONCLUSIONS

In this paper, we proposed an optimized capsule network model for banana leaf disease classification. It is motivated by the newly proposed Capsule Network model which consists of the input layer, convolution layer, primary capsule layer and digitcaps layer. The network was optimized by fine-tuning the hyper parameters and using the Leaky ReLU activation function. The dataset used in this work contains two variants of banana leaf diseases which are banana bacterial wilt disease (*Xanthomonas* wilt) and banana black sigatoka disease. We implemented the variants of CNN architecture (CNN trained from scratch, LeNet5 and ResNet50) in order to compare their performance with our model. The developed model outperformed other variants of CNN architectures implemented with respect to rotation invariance as seen in Table 3 above. However, the test dataset without rotation on ResNet50 architecture gives a better result than the proposed model. This work is limited to the two variants of banana (*Musa Acuminata*) leaf disease and future work includes building a robust capsule network model that can take care of diseases from various plant species.

REFERENCES

- [1] Afshar, P., Mohammadi, A., & Plataniotis, K. N. (2018). Brain Tumor Type Classification via Capsule Networks. *Proceedings - International Conference on Image Processing, ICIP*, 3129–3133. Retrieved from: <https://doi.org/10.1109/ICIP.2018.8451379>
- [2] Amara, J., Bouaziz, B. & Algergawy, A., (2017). A DeepLearning-based Approach for Banana Leaf Diseases Classification. In: Mitschang, B., Nicklas, D., Leymann, F., Schöning, H., Herschel, M., Teubner, J., Härder, T., Kopp, O. & Wieland, M. (Hrsg.), *Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshopband*. Bonn: Gesellschaft für Informatik e.V. (S. 79-88). Retrieved from: <https://www.pdf-s.semanticscholar.org>
- [3] Barbedo, J. G. A. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosystems Engineering*, 172, 84–91. <https://doi.org/10.1016/j.biosystem-seng.2018.05.013>.
- [4] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145(September 2017), 311–318. <https://doi.org/10.1016/j.compag.2018.01.009>.
- [5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (2672-2680).
- [6] Hinton, G. E., Krizhevsky, A., & Wang, S. D. (2011). Transforming autoencoders. In *International Conference on Artificial Neural Networks* 44-51. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21735-7_6.
- [7] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets (2016). *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- [8] Hinton, G. E., Sabour, S., & Frosst, N. (2018). Matrix capsules with EM routing. In *International Conference on Learning Representations, 2018*. <https://openreview.net/forum?id=HJWLFGWrb>.
- [9] Honfo, F. G., Tenkouano, A. & Coulibaly, O. (2011). Banana and plantain-based foods consumption by children and mothers in Cameroon and Southern Nigeria: A comparative study. *African Journal of Food Science* Vol. 5(5), pp. 287 – 291, May 2011
- [10] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
- [11] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. arXiv preprint arXiv:1602.07360.
- [12] Deng, J., Dong, W., Socher, R., Li, L., Kai, L. & Fei-Fei, L. (2009). "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255. doi: 10.1109/CVPR.2009.5206848.
- [13] Kamilaris, A. & Prenafeta, B. F. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*. 147. 10.1016/j.compag.2018.02.016.
- [14] Krizhevsky, Alex & Sutskever, Ilya & E. Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*. 25. DOI:10.1145/3065386.
- [15] Simonyan, K. and Zisserman, A. (2015) Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [16] Kumar, A. D., Engineering, C., Vidyapeetham, A. V., Engineering, C., Vidyapeetham, A. V., Parameswaran, L., & Vidyapeetham, A. V. (n.d.) (2018). Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks. arXiv preprint arXiv:1805.04424.
- [17] Lecun, Y., Bottou, L., Bengio, Y., & Ha, P. (1998). 00043771.Pdf, (November), 1– 46. <https://doi.org/10.1109/5.726791>.
- [18] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7. <https://doi.org/10.3389/fpls.2016.01419>.
- [19] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic Routing Between Capsules, (Nips). <https://doi.org/10.1371/journal.ponea.0035195>
- [20] Sivaranjini, S., & Sujatha, C. M. (2019). "Deep learning based diagnosis of Parkinson's disease using convolutional neural network." *Multimedia Tools and Applications* (2019): 1- 13.
- [21] Xu, Z., Lu, W., Zhang, Q., Yeung, Y., & Chen, X. (2019). Gait recognition based on capsule network. *Journal of Visual Communication and Image Representation*, 59, 159- 167.

- [22] Zhang, S., Huang, W., & Zhang, C. (2019). Three-channel convolutional neural networks for vegetable leaf disease recognition. *Cognitive Systems Research*, 53,31–41. <https://doi.org/10.1016/j.cogsys.2018.04.006>.

AUTHORS

OLADEJO, Olajide Ademola, received his Master's Degree in Computer Science in year 2019 from the Department of Computer Science, University of Ibadan, Ibadan, Nigeria. He is working as a research coordinator at College of Medicine, University of Ibadan, Ibadan, Nigeria. He has published two papers in International Journals and one currently submitted at IEEE. His research interests include Artificial Intelligence, Computer Vision, Digital Health and Machine Learning.



Dr. Bolanle F. OLADEJO is a Senior lecturer at the department of Computer Science, University of Ibadan, Ibadan, Nigeria. She is a recipient of French Government scholarship for a co-supervised doctoral research. She holds a double PhD degree in Computer Science from Nancy 2 Universite (currently University of Lorraine), France and University of Ibadan. Her area of Specialization ranges from Knowledge pitalization/Management, Competitive/Business Intelligence, Software Engineering, and Health Informatics to Information Security. She has published over forty papers from her research outputs in reputable conference proceedings and peer-reviewed journals at both local and international levels. She leads a doctoral research team with bias in medical information visualization, decision support for security in terrorism and application of NLP and Knowledge Discovery techniques to collaborative knowledge sharing.



AN ADAPTIVE UTILIZATION OF CONVOLUTIONAL MATRIX METHODS ON SLICED HIPPOCAMPAL NEURON CELL SEGMENTATION WITH AN APPLICATION INTERFACE

Neeraj Rattehalli¹ and Ishan Jain²

¹Computer Science, Menlo-Atherton High School,
Atherton, California, , USA

²Computer Science, Mission San Jose High School,
Fremont, California, USA

ABSTRACT

Current methods of image analysis and segmentation on hippocampal neuron bodies contain excess and unwanted information like unnecessary noise. In order to clearly analyze each neural stain like DAPI, Cy5, TRITC, FITC and start the segmentation process, it is pertinent to preemptively denoise the data and create masked regions that accurately capture the ROI in these hippocampal regions. Unlike traditional edge detection algorithms like the Canny methods available in OpenCv libraries, we employed a more targeted approach based on pixel color intensities to segment out hippocampal neurons from the background. Using the R, G, and B value thresholds, our algorithm checks if a cell is a boundary point by doing neighboring pixel level comparisons. Combined with a seamless GUI interface for cropping the highlighted ROI, the algorithms efficiently work at creating general outlines of neuron bodies. With user modularity from the various thresholding values, the outlining and denoising presents clean data ready for analysis with object detection algorithms like FRCNN and YOLOv3.

KEYWORDS

Convolutional Matrix, Computer Vision, Machine Learning, Deep Learning, Automation Interface

1. INTRODUCTION

Much of the the hippocampal neuron body image data in the current scientific field contains unwanted data such as noise that is generated. In addition, neuron bodies that have neurons congregated together are unable to be segmented. Multiple neurons in close proximity are identified as a singular body. The current methodologies cause them to be grouped together. Smaller neurons with larger distances among themselves are easier to be segmented. In addition, some neuron data have different colors. Some images may contain a red-dominant color in the RGB spectrum, while others may contain a different dominance, and so forth.

Additionally, current filtration algorithms such as OpenCV's canny edge detection models only function on black and white images [1]. Other existing models such as the Sobel, Prewitt, and

Laplacian edge detection algorithms also leverage a gray-scale conversion while conducting edge detection processes [4, 5, 6]. The processes remove the color from the image, causing the image spectrum to be heavily manipulated. Our goal was to create an accurate model that can perform tasks despite the color gradients present within an image. Hence, our algorithms can process accurately on neuron data without changing the dynamics of the colors of the image.

As evident from current methods, disparities in hippocampal neuron data can cause future complexities for data analysis. When neuron data is close together, most algorithms fail to segment the neurons properly because of the proximities. Because of this, neurons are falsely grouped together with contours, leading to the failure for adaptive segmentation. In addition, unwanted data such as synapses are contoured. Without an effective denoising algorithm, such data will be processed in a further neural network. During the training of the neural network, the synapses will be used as a feature.

Although such issues relating to synapses may be avoidable in the status quo with cropping tools, the scientific field lacks a method that includes a cropping tool and data/image processing algorithms, all in one systematic interface.

Another dilemma in current image analysis methods include the singularity of data that it can process on. Images that contain a dominant color that is unsuitable for the algorithm is rendered useless for current algorithms. The RGB combination demonstrates a dominant color for neuron data. Some neuron stains contain red dominance (TRITC), while other stains such as the FITC stain contains a green dominance. With these disparities, current algorithms are not adaptive. To address this lack of modularity, we utilized user-inputted data that receives a threshold value that compares the image pixel intensities with. With this implementation, the difference in color dominance can be addressed.

With potential problems in current image segmentation techniques, we propose a solution to address all scenarios. Our developed interface includes algorithms that can denoise unwanted data such as synapses, filter through images using convolution methods, and create contours for neuron data despite proximities with each other (disregarding groupings).

2. METHODS

The general methodology for image preprocessing that we employed starts through a web interface which allows for easy upload, labeling, masking, and cropping of ROI. After the images are cropped as targeted for their specific regions, a data.txt file is created containing the relevant information about image size and the specific points on the freehand cropped region. Due to resizing changes in the DOM as compared to original image files, the coordinates of the cropped points must be modulated in accordance to the new dimensions. Once a final correct masked region is created, it is ultimately ready to undergo subsequent denoising and edge detection to create general boundaries around each neuron.

2.1. GUI

Current tools for addressing image segmentation processes include cropping methods. Although images are able to be cropped accurately, current available systematic methods lack the full-stack implementation to create crops, process the region of interest (ROI), and accurately contour the neurons in a single application. Using web development resources, we created a web app that includes a cropping tool keeping UX/UI in mind. With an input-image selector, one can input a given image for processing. After the respective crop is completed, one can download the neuron

data to the local system. By executing a script, the neuron data (preferred threshold, dominant color, etc.) is taken into account, and the image is processed and contoured accurately in approximately 10 seconds on a machine with an Intel Core i5 processing unit. With an application that incorporates a variety of languages (web development and image processing), we developed an application that combines the resources for cropping and filtering/segmenting images in a customizable manner. Fig. 1 demonstrates an accurate representation of the web interface that is employed for the full-stack image segmentation tasks.

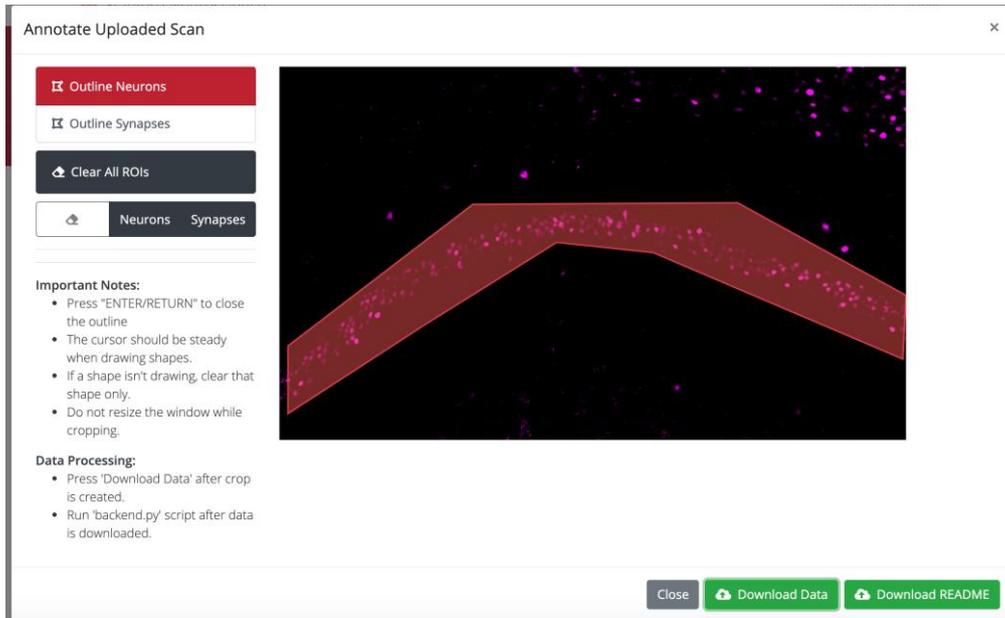


Fig 1. A representation of the cropping tool with a generated crop around the region of interest (ROI). Erasures of different crops can be done via the different options presented.

2.2. Is on Edge

Traditional edge detection algorithms use implemented kernel methodologies to run a convolutional matrix throughout the image to create edges based on pixel intensity contrast. The algorithms present in this paper employ similar convolutional matrix methods however without the use of filtration and substituted with a novel checking system. The algorithm starts by analyzing a 3x3 pixel matrix of the image. The analysis is simply done on the central cell. It starts off by checking the presence of null information which occurs with low pixel intensities for each of the RGB values. For general purposes and through testing, we deemed an ideal value for such a threshold would be 50. Thus, depending on the stain the user is analyzing, the algorithm checks for null intensity on the appropriate RGB value (e.g. the TRITC channel which creates a red stain has a null intensity check for the R value of the pixel). If there exists a pixel in the 3 by 3 matrix other than the center pixel which has a pixel intensity for the respective channel that is less than 50, then it means the center pixel is in the presence of a cell with null intensity. Now, there is a second part for edge detection: it is to determine if the center pixel is in contact with a colored pixel (e.g. for the TRITC channel, if the center pixel is in direct contact with a red pixel). This can be conclusively determined through thresholding. Once again, we deemed an ideal value for this would be 50 as well. Finally, if the center pixel is in contact with both a colored pixel and a pixel with null intensity, then it lies on the edge of a neuron. The general algorithm is presented below in pseudo code.

Line 1: denotes the pixel value intensity for the 3 by 3 cell matrix
 pixels = [[i1, i2, i3],[i4, i5, i6],[i7, i8, i9]]
 // i_n = the pixel color intensity for the appropriate light signal for
 a specific pixel for i = 1,2,3...9.

Line 2, 3: defines threshold limits for null intensity and colored intensity

```
thresholdLower = 50 // Can be set by user thresholdUpper = 50 // Can be
set by user
```

Line 4: function definition for isOnEdge

```
function isOnEdge(pixels):
```

Line 5, 6: boolean variables for states of neighboring pixels
 doesNullIntensityExist = false doesColorIntensityExist = false

Line 7, 8: double for loop to parse through pixels array
 for row in pixels: for pixel in row:

Line 9, 10: Null intensity check
 if pixel < thresholdLower: doesNullIntensityExist = true

Line 11, 12: Color intensity check if pixel >
 thresholdUpper: doesColorIntensityExist = true

Line 13: returns the boolean function for both color and null pixel intensity existing.

```
return doesNullIntensityExist and doesColorIntensityExist
```

2.3. Edge Detection

In the previous section, a general methodology for determining if a certain pixel resided on the edge of a neuron was outlined. In order to actually create a new image with the outlines of edges, this function must be run on each 3 by 3 pixel chunk of the image. If the function returns true, which indicates that the center pixel neighbors both a null intensity pixel and a colored pixel, the center pixel's color is changed to white to indicate that it is on the edge of a neuron. After running this function through each 3 by 3 section, a general edge outlines are created.

2.4. Denoising

After the general edges have been created, the data can be further augmented to relieve some of the additional noise. There are 3 specific ways of doing so: thinning out white borders, strengthening pixel intensities, and removing small regions.

2.4.1. Thinning Out White Borders

Once the edges have been created, there are regions where the border is multiple pixels wide. This starts to take away data from the original image. Thus, it is critical to thin out the white lines as much as possible; this is done in two steps. The first is to eliminate all the white pixels between the inner and outer border of the neuron edge. Essentially, an algorithm checks the 3 by 3 pixel cell matrix around a center cell and checks if all of them are the color of the edge (in our case white). If this is true, the cell color is returned to its original. After this process occurs each neuron is left with 2 borders: an inner one and an outer border. We then proceed to remove the inner border. This is done by simply checking for the presence of null intensity around the center pixel. Null intensity exists around the outer border but not in the inner one. This can eliminate

the inner border. In addition, sometimes there exist lone white cells which are remnants of the thinning out white borders methods. This can also be removed by checking for the existence of null intensity around its neighboring pixels.

2.4.2. Strengthening Pixel Intensities

Once the borders have been thinned out, the image is ready to be further processed. In a lot of the hippocampal data that our lab worked with, there was additional noise present from faint colored synapses. To remove the synapses, a threshold method was used once again. To determine the threshold, we created a bucketing algorithm. The bucketing algorithm grouped all pixels based on their RGB intensity values. It created 10 buckets where each bucket represented intervals of length 25.6. These intervals contained the coordinates of cells with the specific light intensity channel within the end points of the appropriate interval. For example, if one was analyzing the red channel and the specific pixel had an intensity of 14 for its R value, that pixel was bucketed into the interval [0, 25.6]. Since the neuron's and the synapse's intensities differed by at least 50 (this was manually checked), the algorithm looked to find intervals where the quantity of bucketed pixels was smaller than the quantity of bucket pixels than its neighbors. This created an automatic way of determining the threshold pixel intensity value. Then, the image was parsed once more pixel by pixel. If the specific pixel has a lower value than the threshold for the appropriate channel, it was turned to black like the background; this removed a lot of noise. It essentially created a completely black background as opposed to a black background with subtle hints of red.

2.4.3. Removing Small Points

This algorithm is relatively straightforward. It takes 9x9 pixel matrices. The algorithm then proceeds to check if the boundary of these pixel matrices is entirely black. If so, this means that either the entire 9x9 section is black or there were small points that were entirely contained in this 9x9 interval, which need to be removed. If the edge is entirely black, then it blacks out every pixel in this 9x9 pixel matrix since it's either null information or small points. This works because all of the cells in our imaging data were much larger than a 9x9 pixel square matrix.

2.5. Data Transfer

In order to adequately incorporate a method that utilizes a cropping tool with the image analysis algorithms in a systematic full-stack tool, we developed a data transfer method to allow the script to process the image with the crop data that was generated from the web app. After data is downloaded from the web tool, the script processes and parses the input data in a correct format. Then, using our image segmentation algorithms and feature explorations, the image is processed within the same run time. The analyzed image with contours is exported to the local filesystem.

2.6. Data Parameterization

In order to custom fit the algorithms to user specification, the Python terminal interface contains convenient methodology for the addition of the various parameters. The code takes in the input paths to each of the images, a general folder path and naming convention for the various images, and finally thresholding values for each of the various filters. With these thresholding values, the user can custom fit the data augmentation to their needs. The user can determine the ideal threshold value with 8 runs (because there are 256 values for pixel intensity) using a binary search approach.

3. RESULTS

The following (Fig. 2) describes the application of the edge detection algorithm on the Cy5 channel stain. As evident, the white contours are able to distinguish between each neuron and the background.

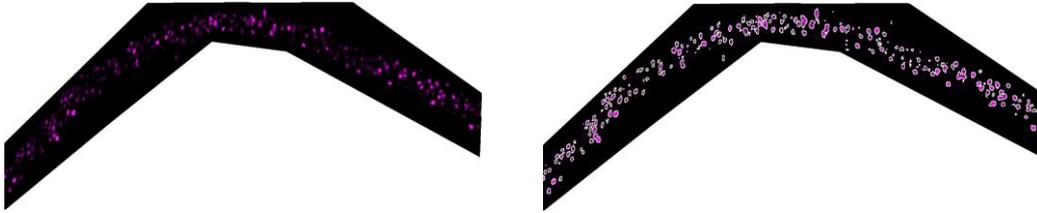


Fig 2. A side-by-side comparison of a before vs. after edge detection process on the Cy5 channel

The following (Fig. 3) describes the application of the edge detection algorithm on the DAPI channel stain. As evident, the white contours are able to distinguish between each neuron and the background.

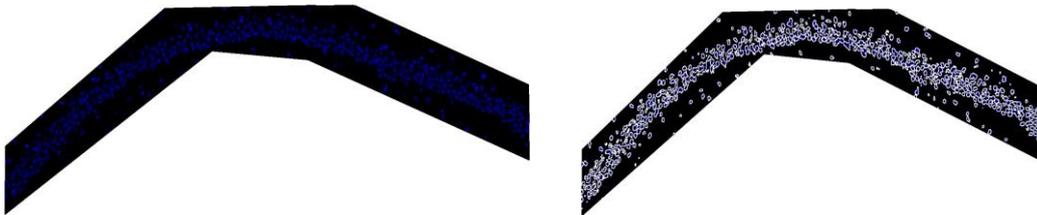


Fig 3. A side-by-side comparison of a before vs. after edge detection process on the DAPI channel

The following (Fig. 4) describes the application of the edge detection algorithm on the TRITC channel stain. As evident, the white contours are able to distinguish between each neuron and the background.

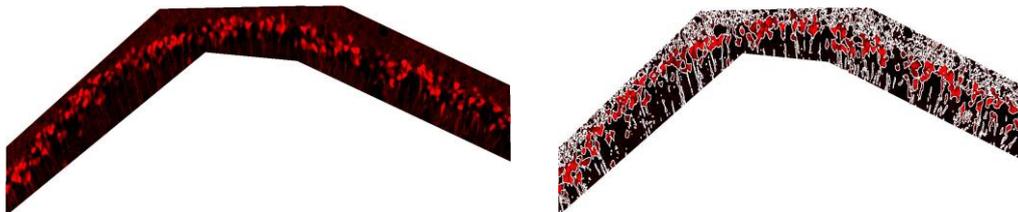


Fig 4. A side-by-side comparison of a before vs. after edge detection process on the TRITC channel

The following (Fig. 5) describes the application of the edge detection algorithm on the FITC channel stain. As evident, the white contours are able to distinguish between each neuron and the background.

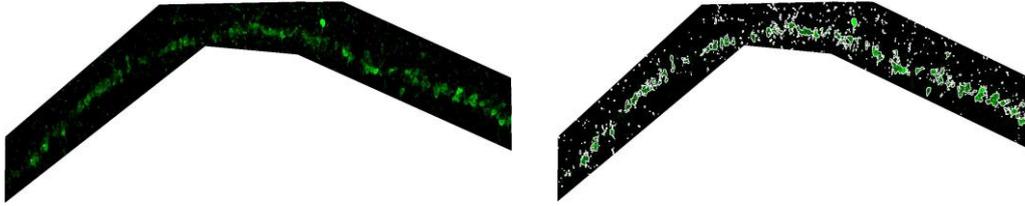


Fig 5. A side-by-side comparison of a before vs. after edge detection process on the FITC channel

The following (Fig. 6) describes the application of the white removal algorithm on the TRITC channel stain. As exhibited, the density of the white contours has been reduced.



Fig 6. A side-by-side comparison of a before vs. after White Removal algorithm process on the TRITC channel

The following (Fig. 7) describes the application of the small point removal algorithm on the TRITC channel stain. As exhibited, the noise due to the synapses (small points) has been reduced. The final image at the bottom demonstrates the small point removal algorithm with the crop applied which has crisper quality as opposed to the original data.

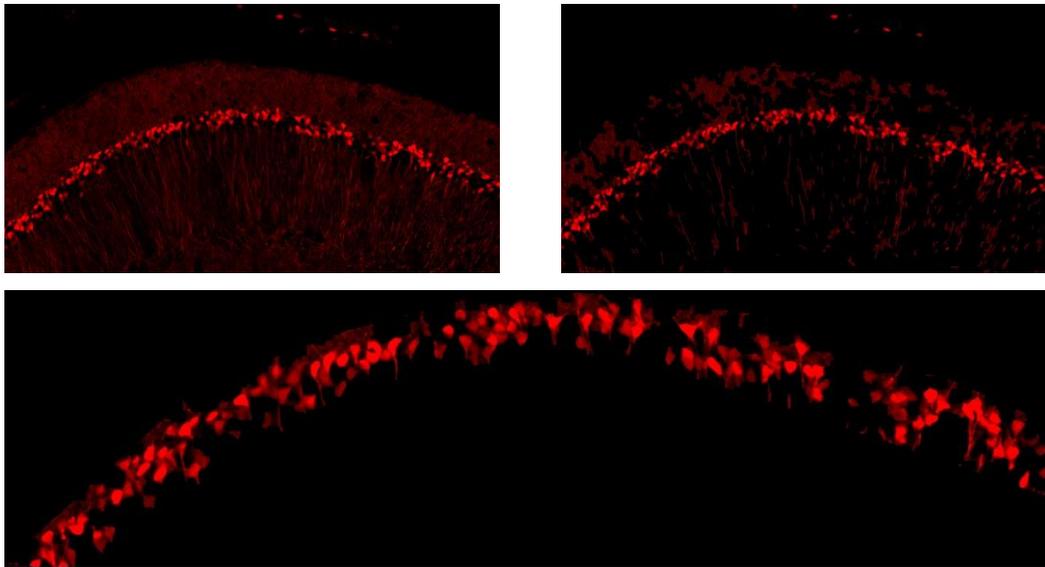


Fig 7. Top: A side-by-side comparison of a before vs. after Small Points Removal algorithm process on the TRITC channel

Fig 7. Bottom: Small Point Removal algorithm process on the TRITC channel With Crop

The following (Fig. 8) describes the application of the strengthen pixel intensity algorithm on the TRITC channel stain. Paired with the small point removal algorithm, this final data is the most

clean as it contains thin borders/contours, an accurate crop, and little to none noise due to the synapses.



Fig 8. A Side-By-Side Comparison of A Before and After of the Strengthen Intensity Algorithm on TRITC Channel

4. CONCLUSIONS

In summary, this study demonstrates an algorithmic architecture that is capable of automating the segmentation of neuron data on various stains experimented on different hippocampus slices. In addition, the use of an interface to allow researchers and scientists to input image data in a robust method exhibited the increase in image segmentation techniques in the field of machine learning and automation. This approach provides sufficient resources for further analysis on images, for the contoured neurons can act as training data for future training or analysis.

ACKNOWLEDGEMENTS

We would like to thank our mentors, Professor Lu Chen and Dr. Bing Cao for guiding us through our project and helping us with the sufficient resources. In addition, we would like to thank Dr. Jie Li for providing us with our training and input data. We thank Dr. Amy Shon for providing us with access to GPU and computing resources such as the Sherlock cluster for us to train and run our models and matrices. Finally, we thank the Lu Chen lab from Stanford for providing us with the images and data that was adequate for conducting our research.

REFERENCES

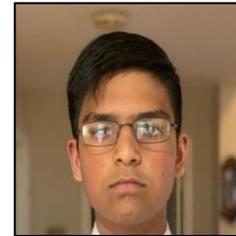
- [1] J. Canny, "A Computational Approach to Edge Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.
- [2] I I. Culjak, D. Abram, T. Pribanic, H. Dzapov and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012, pp. 1725-1730.
- [3] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788
- [4] Dey, D. and Polley, D., 2020. Edge Detection By Using Canny And Prewitt. [online] Ijser.org. Available at: <<https://www.ijser.org/researchpaper/Edge-Detection-by-Using-Canny-and-Prewitt.pdf>> [Accessed 25 June 2020].
- [5] Owl.net.rice.edu. 2020. Laplacian Edge Detection. [online] Available at: <<https://www.owl.net.rice.edu/~elec539/Projects97/morphjrks/laplacian.html>>

AUTHORS

Neeraj Rattehalli is a Bay Area high school student with a passion to automate the modern world. Growing up in a rapidly changing progressive society, Rattehalli has found machine learning and computational biology around every corner, and in order to stay steps ahead of the crowd, Rattehalli is conducting novel research at Stanford under the guidance of Lu Chen, Professor of Neuroscience. Rattehalli works with the intention and the zeal to solve today's most intricate challenges.



Ishan Jain is a Bay Area high school student with a passion in machine learning and computer vision. Ishan has worked on a variety of projects at Stanford Medicine, including analytical methods for assessing patients with peripheral artery disease (PAD) and developing mobile tools to create a remote surveillance application for postoperative surgery treatment using opioid medications. Furthermore, Ishan has worked with research professionals at the University of California, Santa Barbara, where he developed a meta-analysis tool to automate research manuscripts. Ishan is highly passionate about utilizing computer vision methods in application to the modern world.



© 2020 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

WIRELESS SENSOR NETWORKS SIMULATORS AND TESTBEDS

Souhila Silmi^{1,2}, Zouina Doukha¹, Rebiha Kemcha^{2,3}
and Samira Moussaoui¹

¹Department of Computer Science, USTHB University, RIMAA Lab., B P
N°32 El Alia, 16000 Bab Ezzouar, Algiers, Algeria

²Department of Computer Science, Higher Normal School Elbachir El-
ibrahimi-Kouba, B P N°92 16308 Vieux-Kouba, Algiers, Algeria

³Department of Computer Science, University of Boumerdes, LIMOSE
Laboratory, Boumerdes, Algeria

ABSTRACT

Wireless sensor networks (WSNs) have emerged as one of the most promising technologies for the current era. Researchers have studied them for several years ago, but more work still needed to be made since open opportunities to integrate new technologies are added to this field. One challenging task is WSN deployment. Yet, this is done by real deployment with testbeds platforms or by simulation tools when real deployment could be costly and time-consuming. In this paper, we review the implementation and evaluation process in WSNs. We then describe relevant testbeds and simulation tools, and their features. Lastly, we conduct an experimentation study using these testbeds and simulations to highlight their pro and cons. As a use case, we implement a localization protocol. This work gives clarity to future-work for better implementation in order to improve reliability, accuracy and time consumed.

KEYWORDS

Wireless Sensor Network, Testbeds, Simulation Tools, Localization Protocol

1. INTRODUCTION

The continuous evolution of the need for study and performance evaluation of complex applications, such as security, supervision, military applications, medical and environmental applications, has given importance and necessity to several tools for implementation of Wireless Sensor Networks (WSNs). Recently, WSNs monitor our cities and living environments [1]. With the advent of new technologies, WSNs are integrated with other emerging technologies, giving birth to new architectures such as Vehicular Ad-hoc Networks, Hybrid Sensor and Vehicular Networks (HSVN) , Internet of Things (IoT), and Smart Cities. However, WSNs are difficult to deploy for many reasons. The complexity of the environment, in which the system operates leads to many challenging issues for the designers [1]. The need for adequacy study of the proposed solutions to the reality motivated the development of various tools of test and implementation for new protocols. For that, several tools are available namely: analytical methods, simulations tools, emulation, and prototype generation. In fact, after the use of these implementation methods, a risk of imperfection towards the reality remains. Actually, it is common to estimate the protocol in a real and concrete context. In this direction, adapted tools emerge, not only to make it more accessible task, but also to master the deployment of these networks.

In several research institutions, testbeds are proposed for various research experiences. Typically, they consist of sensor nodes deployed in a controlled environment, and provide a platform for experimenting with large projects. Research on WSNs is highly expensive when real-time sensors are deployed in specific environment due to constraints on complex topology, and the area of deployment. Thanks to these testbeds, large-scale of resources becomes available. This challenge requires the implementation of simulation environment, with regard to real conditions. In WSN, the availability of multiple simulation tools, making the choice of researchers more difficult to take. In this study, we are interested in a comparative study between the relevant testbeds and simulators in the literature in view of criteria as processing capabilities, accessibility, types of available tests and reliability of results. To the best of our knowledge, there is no comparative study, in the literature, that includes both simulators and testbeds with specific use case.

This paper is divided into six parts: the second section summarizes the existing studies in the field of WSNs implementation. Section 3 presents the simulation tools, their characteristics and practical remarks about the studied simulators. Then, in section 4, we describe the different testbeds in WSNs where we give our remarks for each testbeds studied. Section 5 defines the localization function in WSNs as a use case for our experiments and gives our results with different approaches of implementation. Finally, in section 6, we conclude and present our research open issues that need to be investigated in future work.

2. RELATED WORK

The protocols engineering implies numerous phases of tests to validate new solutions. In the specific domain of WSNs, experiences are often complex, hard to repeat and slow to configure and execute. For these reasons, the simulation was the methodology widely used by researchers. However, the researchers become more and more conscious of the fact that the current simulators are unable to model certain essential characteristics of the real systems. For this reason, and due to a visible degradation of specific standards in the driving simulation studies, the simulation results are often debatable and subject of credibility [2]. Therefore, we are aware of a fundamental importance to take forward the theoretical design. The analysis of protocols and algorithms serve in a parallel to the experimental validation, by the use of simulators tools. For such a purpose, the testbeds experiments are proposed.

Several surveys exist in the literature on different research objectives. Some of them provide a comparative study between testbeds. In [3], supporting standards, storage and physical architecture for three platforms are given. A resume about some simulators, with a comparative table about the five highly evaluated tools is given in [4]. In order to explore synergy, authors in [5] review prominent research projects for the different standards and technologies used by platforms. Before giving some detail about testbed, we can find in [6], general information about how to prepare and exploit experimentations in several fields. This survey proposes divers characteristics of protocols for IoT and WSN. It discusses features such as heterogeneity and scaling for the material aspect. Egea-López and all, in [7], study the basic properties to select the appropriate simulation environment. In [8] a comparative study shows that simulators, with regard to VANETs environment, must revise more characteristics. We find works that propose new testbed/simulator, or new architectures. For instance, the authors of [9] intend to give a description of a flexible and non-specific management system, unlike the current management systems, which are strongly coupled to a specific testbed configuration. After providing a summary of the best-known testbeds in [10], this paper proposes a new data recovery mechanism that aids to store the results of structured data in this testbed. This article gives mainly the software aspects of the work. Others propositions are given in [11]-[13].

This paper addresses the comparison between the most popular simulators for WSNs and a comparison between relevant testbeds. A use case is presented to study the effectiveness of two simulators, and two testbeds.

3. THE SIMULATION

The simulation of network is definitely one of the most dominant evaluation methods in the field of networks. It is widely used for the development of new communication architectures and network protocols. A simulator is software that imitates the behavior real world components and is used as a research and development tool. Depending on the intended use of the simulator, different parts of the system are modeled. Previous simulators designed for WSNs model the wireless transmission in the network, but currently sensor network simulators have a more detailed model and realistic, including barriers between nodes [14]. A more abstract model is given by the recent simulation tools to get closer to real environments.

3.1. Necessity of the Simulation

The emergence of WSNs has brought new problems to network designers. Computer simulations, analytical methods, or physical measurement (testbeds) can achieve the evaluation of networks performance. Real experiments are important to network research. Although they use complex coding and laboratory experiments, they are able to give important details automatically. However, this approach has disadvantages; real experiments are more expensive to build, and adapting the configuration of laboratory scenarios can be difficult. This leads to limited power sharing and flexibility.

Another point, which can make the evaluation and comparison of protocol designs even harder, is the difficulty to reproduce experimentally as some networking phenomena such as wireless radio interference. The algorithms complexity, in WSNs, is related to their constraints such as limitations due to energy, fault tolerance and decentralized collaboration. Therefore, it appears that the simulation approach is the most feasible approach for quantitative analysis of sensor networks before a real deployment [15]. Simulation allows doing the tests at lower cost and making important decisions.

3.2. Simulation Tools

3.2.1. Network Simulator (NS-2)

NS-2 [16] is one of the most widely used simulators in research laboratories to simulate and study the performance of network protocols. It was developed using discrete event technique for network research. It started as a variant of the REAL network simulator [17] in 1989 and evolved considerably nowadays [4]. It has a modular approach and the simulations are based on a combination of C++ and Otcl languages. NS-2 offers the opportunity to test, analyse and evaluate applications before considering the practical implementation in real networks. NS-2 simulator, particularly, is well suited for packet switching networks and large scale. It contains the necessary functionalities for the study of unicast or multicast routing algorithms, transport, session, booking protocols, integrated services and localization protocols. NS-2 use Network Animator (NAM) as graphical visualization tool.

3.2.2. NS-3 Simulator

The NS-3 project, started in 2006, like an open-source project developing NS-3. The NS-3 Simulator is a discrete-event network simulator oriented mainly for research and educational use [18]. Unlike its predecessor NS-2, NS-3 Simulator is based only on C++ language for the implementation of simulation models. Thus avoiding problems caused by the combination C++ and oTcl in NS-2. Therefore, network simulations can run on a purely C++ environment, while NS-2 users can possibly run simulations using Python as well [19]. In addition, NS-3 integrates architectural concepts and the GTNetS Simulator code that has good scalability features. The new features appeared with NS3, were made at the expense of compatibility. In fact, NS-2 models have to be brought to NS-3 manually. Several external animators and visualization tools can be used with NS-3 like NS-3-Viz, PyViz and NetAnim [18].

3.2.3. Tossim Simulator

Tossim is a simulator for TinyOs operating system, and it was developed at UC Berkeley [20]. It simulates the behavior of a sensor (sending/receiving messages via radio waves and processing information). This simulator is written in NesC Language which provides a component-base programming model. TOSSIM simulates the TinyOs network stack at bit level, allowing experimentation with low-level protocols in addition to high-level application systems. TOSSIM is based on the assumption that each node in the network must run exactly the same code, which makes it less flexible. TOSSIM does not model energy consumption, so PowerTOSSIM Simulator and PowerTOSSIM z are an improvement that extend the simulator to model energy consumption [21].

3.2.4. OMNet++ Simulator

Unlike ns-2 and ns-3, OMNet++ is not a network simulator by definition, but a discrete framework of general use based on simulation events. Although it is most often applied to the field of network simulation, given that its INET package offers a complete collection of Internet Protocol models. In addition, it has other models packages like the mobility package for mobile Ad-Hoc networks and mobile WSNs [18].

3.2.5. Avrora Simulator

It is an open source simulator for embedded detection programs. The current version (version 1.7.106) is written in Java. It models two typical platforms, Mica2 and MicaZ [22]. It also provides a framework for program analysis, static verification of embedded software and infrastructure for future programs. Avrora simulator is flexible providing a JAVA API to develop analytics [23] [24].

Table 1. Comparative table of simulation tools in WSNs.

Simulators	Language	Main characteristics	Limits
NS-2	C++ / OTcl	+ Easy to add new protocols + A large number of publicly available protocols + Has a visualization tool + Has a rich collection of models	- Designed for wired networks basically - Uses pure source code in terminal for the development of the simulation set
NS-3	C++	+ Perform large-scale network simulations efficiently [18] + Low calculation and memory requests + Has a visualization tool	-Difficult to implement and simulate a protocol than NS-2 [18]
TOSSIM	NesC (dialect of C)	+ High degree of accuracy by using a few low-level components models [24] + Has a visualization tool	- Simulates only TinyOS programs [20] - Can only emulate the homogeneous applications for the same type of sensors
OMNet++	C++	+ Perform large-scale network simulations efficiently + Has a visualization tool + Has an abstract modelling language	- Unavailability of several important protocols - The combination of models may be difficult and programs are more likely to have bugs [18]
AVRORA	Java	+ An instruction-level sensor network simulator + Has TOSSIM performance while preserving cycle accuracy [24] + Portable due to use of Java virtual machine	- Does not have network communication tools - Does not have GUI - Cannot simulate network management algorithms [24]

3.3. From Simulation Tools to Experimental Testbeds in WSNs

The study shows that each simulator has strengths and limitations, as shown in Table 1. However, the challenge of development, deployment and debugging of applications in realistic environment may not be satisfied with simulation. Many current simulators are unable to model several essential features of the real world. Simulation results are only as good as the model then are only estimated results. Especially for simulation models of WSNs which do not capture the irregularity in the network, either for radio irregularity on the communication [25]; or the irregular sensor model [26], for example. Recent material spread has allowed for new uses for protocol designers. Actually, after the classical phases of specification and simulation, it is common to evaluate a protocol in a real and concrete context. So, adaptive tools are emerging, not only to make this task more accessible, but also to master its deployment, in order to have more usable results.

4. TESTBEDS FOR WIRELESS SENSOR NETWORKS

During our research, we found several testbeds. We have made a choice for the most significant ones and we highlight their characteristics to make it easy for researcher's investigations.

4.1. W-iLab.t

W-iLab.t is a Wireless testbed that enables to test wireless protocols or applications in a real-time environment [27]. The IBBT iLab.t technology Centre manages this testbed. It is based on MoteLab testbed, and is located in Ghent, Belgium. W-iLab.t testbed has a web interface to plan, download, monitor the experience and recover the results. Only authorized users can access the interface. To enable mixed Wi-Fi node and sensor node experiments and to keep a uniform interface, they integrate the support for the Wi-Fi nodes into the same web interface as used for the sensor nodes [28]. Unique features of the test include real-time monitoring of energy consumption and emulation of battery capacity. W-iLab.t has various types of wireless nodes, which are connected to a wired interface for management, and then they can be used during experiments as a cable interface. Heterogeneous experiments wireless/wired are possible. Experimenters are able to give the complete configuration for each device in the network. With this testbed, measurement and management data are saved in a database. This data is made available to the user to be exploited either for processing or for only visualization.

4.2. Tutornet

Tutornet [29] is a testbed for WSNs at Southern California University. It has three levels; test servers, bridge stations and sensor nodes. This testbed includes 104 sensor nodes (91 TmoteSky and 13 MicaZ). The nodes are attached to the gateway stations via USB connections. A Star-gate with several nodes around it form a cluster. Currently there are 13 clusters and the nodes can be programmed distantly.

4.3. MoteLab

MoteLab [30] is a testbed for WSNs at Harvard University. It is accessible for the development and testing of sensor network applications via a web interface. Registered users can download and associate executable files to nodes, to create and plan an experiment to run. MoteLab aims to facilitate research in programming sensor network environments, communication protocols, system design and applications. This testbed contains 190 TMote Sky sensor nodes. The motes consist of the MSP430 processor operating at 8MHz, 10KB RAM, 1Mbit flash memory and a Chipcon CC2420 radio operating at 2.4GHz with an internal range of 100 meters. Light, temperature and humidity sensors are integrated in each node. Nodes run the TinyOS operating system and use the NesC language.

4.4. Wisebed

Wisebed [31] is an experimentation platform in WSN for research purposes. This platform is distributed on nine geographically dispersed sites; it makes available to users more than 750 heterogeneous sensors. Once the registration is confirmed, the user has access to start the configuration of the nodes, as it is depicted in Figure 1.

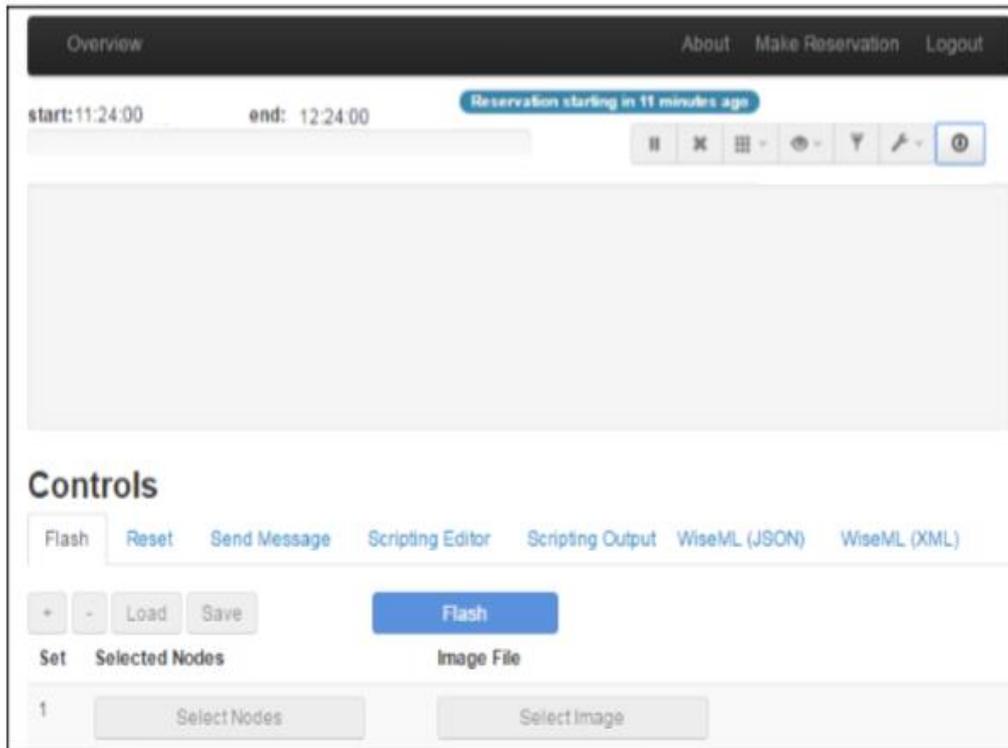


Figure 1. Experiment in WISEBED testbed.

4.5. FIT IoT-LAB

FIT IoT-LAB (IoT-LAB) is a proposed testbed to assure a large infrastructure for scientists in WSNs field. Heterogeneous communicating objects and small wireless sensor devices can be easily tested by this platform. It is an improvement of the SENSLAB testbed (2010-2013) [32], which is located at six different sites in France, with 2728 heterogeneous nodes. IoT-LAB testbed is a well designed system so that user can specify needed properties like the location, radio chip and whether nodes are mobile or not. It allows the repeatability of experiments [6].

4.6. Nitos

The Network Implementation Testbed (NITOS) is developed by NITLAB (Network Implementation Testbed Laboratory) using open source platforms at University of Thessaly in Greece. The NITOS installation currently includes more than 100 operational wireless nodes and is designed to ensure reproducibility of experimentation while supporting the evaluation of protocols and applications in real-world environments. This platform offers testbeds for several types of networks technology [33]. The NITOS testbed includes three different testbeds: the NITOS outdoor testbed, the NITOS indoor testbed, and the NITOS office testbed. All these testbeds are connected with a backbone connection provided by the Greek NREN. They use a part of the pan-European GEANT network [34]. The NITOS testbed is open to all researchers wishing to test their protocols in real-world environments. They have the opportunity to implement their new protocols and study their behavior in a parameter-able environment.

4.7. Planet Lab

Planet Lab [35] is an open platform, shared on a large scale to conduct real world experiments to develop new network services and technologies. It is developed in March 2002 by a network of international private and public laboratories [36]. Although Planet Lab is not dedicated to sensor networks, it is associated with different types of testbeds, including testing on WSNs as part of the initiative to develop testbeds for future Internet. This platform comprises currently more than 1000 nodes on 500 sites throughout the world including Asia, USA, Europe, and Japan. There is also OneLab testbed initiative, which extends Planet Lab Europe by uniting it with other Planet Lab testbeds around the world as well as other types of tests.

4.8. ORBIT

The ORBIT project provides a flexible wireless network testbed open to the experimental researchers [37]. It is designed to achieve reproducible experimentation while supporting realistic evaluation of protocols and applications. ORBIT was funded in 2003 under the Network Research Testbeds (NRT) program [11]. Figure 2 shows the ORBIT radio grid, which was first released to researchers. Currently, it is heavily used when evaluating architectures and protocols in wireless networks. In this testbed, tests on nodes-sensors is given via the Linux terminal [38].

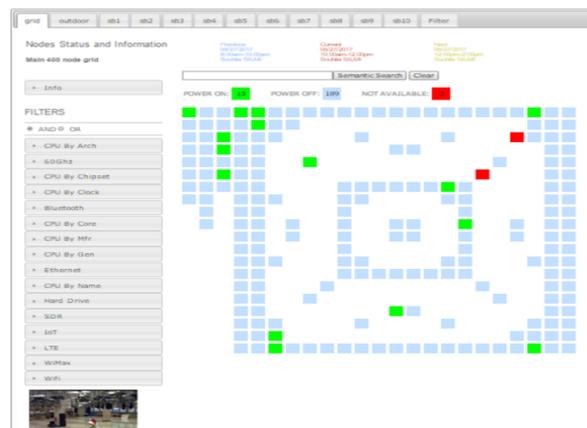


Figure 2. Radio Grid testbed for ORBIT

4.9. Twist

TWIST [39] is an evolutionary and flexible testbed, which is developed by the Telecommunications Network Group (TKN) at Technical University Berlin. It supports heterogeneous hardware and deploys 204 nodes (102 TmoteSky nodes and 102 eyesIFXv2 nodes). Twist testbed allows testing algorithms on different platforms and provides basic services such as node configuration, network programming, off-band retrieval of debugging data and application data collection, as well as several specific functionalities. The lowest level of its architecture is connected by USB cabling and USB hubs to the test infrastructure. The second level contains a «super nodes», which can interact with the USB infrastructure of first level; and the last level has the server and control stations that interact with the super nodes using the testbed backbone. The server stores information on registered nodes in its database and provides remote access via a web interface as depicted in Figure 3. This testbed supports different communication protocols such as Wi-Fi, Zigbee and Bluetooth. Tests on the reserved sensors can be done either via the graphical interface (called Jfed interface) (like for W-ilab.t2 testbed), or via the Linux terminal (like for Orbit testbed).

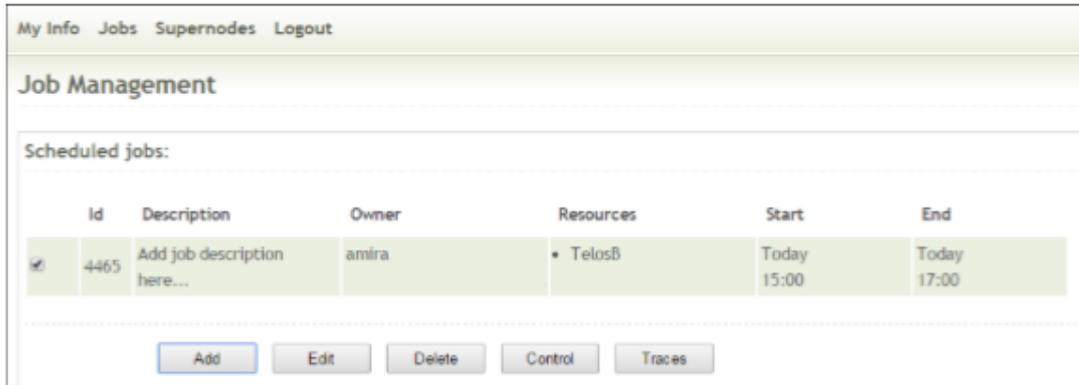


Figure 3. Twist reservation of resource manager

4.10. CERIST

CERIST testbed is developed by the Research Center for Scientific and Technical Information in Algeria. It has more than 30 sensor nodes (10 Telosb and 21 MicaZ). Figure 4 shows the testbed environment.

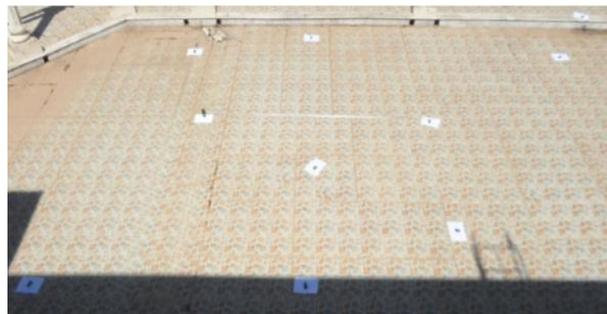


Figure 4. Sensors positioning in CERIST testbed.

Table 2. Comparative table for testbeds in WSNs.

Testbed	Authorized Population	OS	Sensor type	Advantages	Limits
W-ilab.t	Open access	TinyOS	Zolertia Z1	+ Graphic interface. + Accessible via EmuLab portal [40]	- Competition of reservation. -Request to the authority IMinds
TutorNet	Open access	TinyOS	91 Tmote Sky 13 MicaZ		- Incorrect site link
MoteLab	Open access	TinyOS	TMote Sky	+ Graphic interface. +Uses database	-No Web portal. - Incorrect site link
NITOS	Open access	Contiki	M3/A8 tiny NITOS mote	+Heterogenous testbed. +Accessible via OneLab portal	- Competition of reservation.
IoT-LAB	For members	Contiki FreeRTOS	Msn430, M3,M8	+Many platforms. +Accessible via OneLab portal.	- Director's agreement is required for

Testbed	Authorized Population	OS	Sensor type	Advantages	Limits
		TinyOs Riot, OpenW sn			students
Orbit	For members	TinyOS	Tmotestelosb	+Comfortable interface.	-Not scalable -Competition for reservation.
PlanetLab	For members	TinyOS		+ Compatible applications with Linux and Fedora8 + Accessible via OneLab portal.	- Competition of reservation.
Twist	Open access	TinyOS	Tmote Sky eyesIFXv2	+Experimentation on different platforms. +Heterogenous testbed.	- Does not detect automatically the connected nodes [39]
CERIST	Open access	TinyOS	Micaz / Telosb	+ Personalisable environment	-No web portal -Not scalable

4.11. Discussion

Although testbeds are diverse and their services are rich and attractive, they are still subject to credibility. Table 2 summarizes our results and remarks about relevant studied testbeds.

As remote interaction with resources is always dependent on the reliability of internet connection, several times, we needed to redo the whole procedure of preparing the environment and test files due to the interruption of the connection to the web interface.

Account creation seems easy and instantaneous, but validation access can last longer. Most testbeds require selecting the organization to which the user belongs; otherwise, access is not possible for foreign students (Tutornet, OneLab and MoteLab cases).

Sometimes, the resource reservation system encounters some access contention, and this can cause the interruption or the cessation of experiments (ORBIT case).

Maintenance of websites without prevention (WISEBED case) and the greater level of access concurrence makes the available hours too limited, and it may be impossible to complete all tests (W-iLab.t case, when our reservation is not achieved due to unavailability of service with maintenance works). Further, the exhaustion of time reservation implies the immediate disconnection of the server, which can cause data loss and discontinuation of tests.

Most testbeds provide web interfaces for task planning (FIT IoT-LAB and WISEBED cases).

To manage requests, IoT-LAB and WISEBED testbeds use a first come first serve service, which means that the first user requesting experience on available resources gets access.

Testbeds tutorials are not comprehensive enough and explanatory (TWIST case, when access to the sensors in finale step requires a password that was provided to us by the administrators and TWIST tutorials do not mention it).

5. PROTOCOL IMPLEMENTATION IN WSNS

5.1. Localization Protocol as a Use Case

Among the fundamental functionality studied in WSNS, one is the localization of sensors. In the literature, several systems and algorithms deal with this problem. However, research studies aim to develop cost-effective and self-localization algorithms in order to improve the performances. The existing solutions can be classified according to the adopted approach as twofold: The fine localization and the approximate localization. The fine localization approaches accurately determine the coordinates of each node while coarse localization approaches give non-precise position. For the approximate localization approaches, node will use other techniques to estimate an approximate localization. In this section, we study the performances of existing protocols, with regard to simulation and testbed implementation, to determine the gap between the obtained results and see how much the results are reliable.

Heurtefeux and Valois in [41] propose a Qualitative Location Protocol (QLoP). It is considered as one of the most powerful protocols that gives approximate localization for sensor nodes. They use local connectivity information to allow every node to determine the proximity coefficient of its neighbors in one hop [42]. QLoP protocol considers a qualitative metric to describe neighbors like very close neighbor, close or far.

5.2. Implementation with Simulation

In QLoP Protocol evaluation work, Avrora and NS2 simulators conducted our simulation tests.

5.2.1. Operating System and Programing Language

There are several operating systems for sensor networks such as TinyOS, Contiki, Mantis OS, Nut/OS and SOS. TinyOs is the most famous and most complete for several types of sensors like the Mica family, the Telos family and the Iris family. In our simulations, we used TinyOs sytem with NesC language. NesC language has a component-based architecture. The implementation of components is done by declaring tasks, commands or events.

5.2.2. Simulation Environment Parameters and Metrics

The static nodes were placed regularly on a grid in a surface of 100 * 100 meters. Table 3 shows our simulation parameters, as the transmission range (R) is used to control the average degree of nodes in the network. The number of well-located nodes is calculated as one of the metrics, which estimate the accuracy.

Table 3. Simulation parameters.

Paramter	Value
Nodes number (N)	100
Communication range (R)	10, 15, 20 meters
Sensor type	MicaZ
Topology	Static
Simulator	Avrora, NS2

5.2.3. Simulation Results

Figure 5 shows the simulation results by applying different communication range. The green nodes, in the figure, represent nodes that classify correctly their neighbors, while black nodes do not classify correctly their neighbors. For nodes on the edges of the simulation space, the algorithms cannot order correctly the neighbors of the first two proximity classes, due to inconsistencies in the neighborhood.

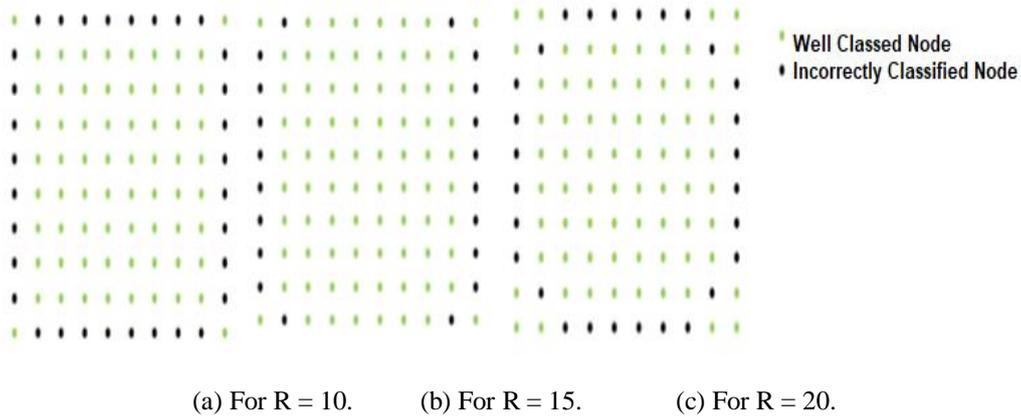


Figure 5. Simulation Results.

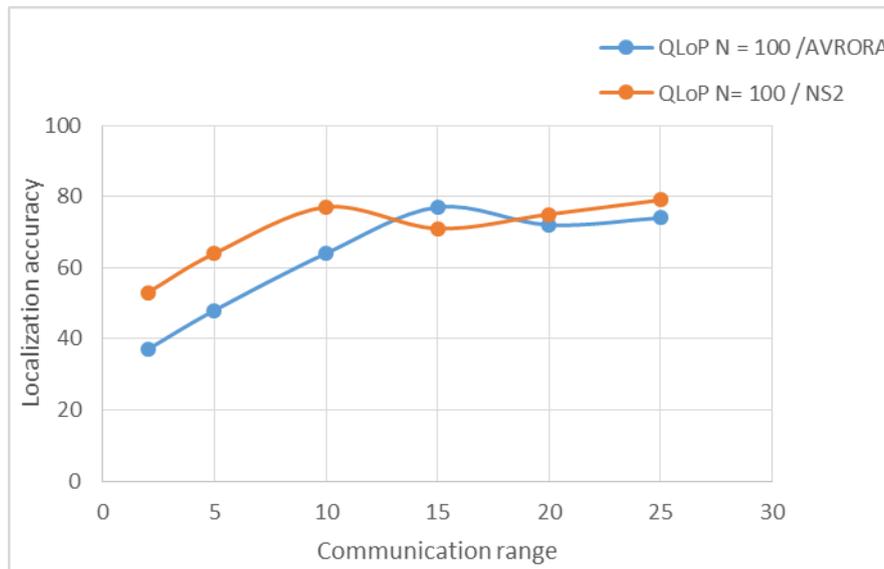


Figure 6. Localization accuracy according to communication range with NS2 Simulator and AVRORA Simulator.

Figure 6 shows the localization accuracy with NS2 Simulator and AVRORA Simulator. The two curves have the same general speed, although NS2 results for communication range less than 5, starts from an accuracy around 50, contrary to AVRORA results which begins with an accuracy less than 40. In [41], authors prove with simulation that the localization accuracy is more than 80 for different communication ranges. Then, NS2 Simulator gave more closed results than AVRORA Simulator.

5.3. Implementation with CERIST Testbed

In our real experimentations, we used CERIST testbed. Initially, this testbed worked with Emulab testbed. However, the access to these resources is interrupted [43]. We visited this research center to use the remaining hardware of this platform. We implemented the protocol using two types of sensors: Micaz and Telosb sensors, in collaboration with sensor team of the Center. In our implementation, we used 10 MicaZ and 5 Telosb, as shown in Figure 7.

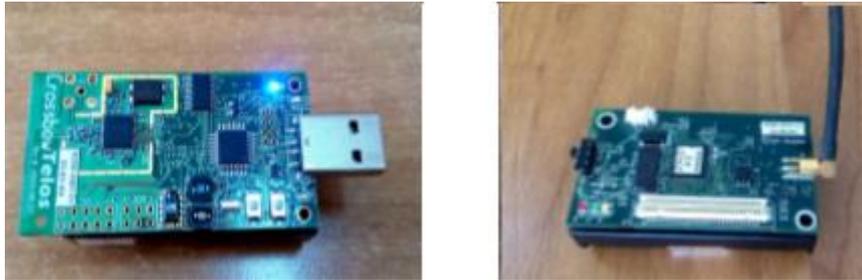


Figure 7. CERIST testbed sensors.

The platform is set up in two parts: software installation and hardware installation. For the software part, we used NesC language with TinyOs system, in Linux environment (Ubuntu distribution). It is possible to use windows environment but there are problems with the second version of Tinyos (Tinyos-2-x) in this environment. We follow many steps to install Tinyos2.1.2, Nesc and micro controller MSP430 for the Telosb and Micaz sensors.



Figure 8. Platform Components.

For the hardware installation, we used a base station connected to the computer via USB, and different sensors like sender / receiver, as shown in Figure 8. Each sensor launches the localization project to classify its neighbors. Then, it sends out the results and makes them available offline in the base station. All sensors communicate via a wireless connection and the base station communicates with the computer via USB.

5.3.1. Implementation Topologies

A network is a set of inter-connected terminals to share information. In our experiments, we used two types of topologies: to reflect reality. The nodes are placed randomly, in the random topology. Nevertheless, the uniform topologies have the advantage of being simple to visualize, then we used also a grid topology. Figure 9 shows an example for each kind of topology.



Figure 9. Random Topology and a grid Topology of deployment.

5.3.2. Nodes Placement

How to place the nodes in the area of interest is an important factor for the localization problem. Among the placement types, there exist:

- Uniform placement: Example of such placement is the grid. However, it is not suitable for dense network;
- Dense placement: which has cost and signal interference problems;
- Incremental placement: In this type, global calculations are made to estimate the optimal place for each new sensor. This mechanism is recommended for WSNs. The incremental placement needs adaptation after installation to fit the communication requirements;
- Self-deployment placement: It maximizes network coverage and takes into account the problem of obstacles. The nodes must perform self-deployment placement autonomously.



(a) Random Topology. (b) a grid Topology of deployment.

Figure 10. CERIST testbed deployment.

5.3.3. Implementation Results

The network can be implemented with homogeneous or heterogeneous sensor-nodes. With the first, all sensors have the same characteristics (like operating range, sensitivity, finesse). However, the other heterogeneous networks are multi-supplier networks, so the hardware or software components come from different suppliers. In our case, we used 10 Telosb and 5 Micaz, like shown in Figure 10, for a random topology or a grid topology. We used two topologies with the heterogeneous sensor-nodes, and the rests were homogeneous ones.

The metrics used in real implementations are the same used in simulations. The 'Results' column in next tables contains two results: Right and Wrong. 'Right' result represents the node that has

correctly classified its neighbor. For example: in table 4, node 1 has correctly classified its neighbor node 4, as its classification is 3 (far class). This classification is equal to the real class. For $N = 4$, almost all nodes do not correctly locate their neighbors. We have identical results for both topologies.

Table 4. Implementation results for a grid and random topologies for $N = 4$ nodes.

Node	Neighbors	Distance (m)	Transmission range (R)	A grid result	Random result	Real class	Note
1	2	2	3	1	1	2	Wrong
1	3	2	3	3	3	2	Wrong
1	4	2.82	3	3	3	3	Right
2	1	2	3	1	1	2	Wrong
2	3	2.82	3	3	3	3	Right
2	4	2	3	3	3	2	Wrong
3	1	2	3	1	1	2	Wrong
3	2	2.82	3	1	1	3	Wrong
4	1	2.82	3	1	1	3	Wrong
4	2	2	3	1	1	2	Wrong

Table 5 and 6 represent the implementation results of QloP protocol for six nodes. For more nodes in the network, we have better results for both topologies. Really, for a random placement of nodes, six nodes could correctly class their neighbors. While in a regular placement, with a grid, we have only four nodes that could correctly class their neighbors. The results were the same for $N= 9$ and $N =12$. When the number of nodes is increased, the results improve more and more.

Table 5. Implementation results for a grid topology for $N = 6$ nodes.

Node	Neighbors	Distance (m)	Implementa tion result	Real class	Communicati on range (R)	Result
1	2	2	2	2	3	Right
1	3	2	1	2	3	Wrong
1	4	2.82	2	3	3	Wrong
1	5	4	1	3	3	Wrong
2	1	2	1	2	3	Wrong
2	3	2.82	3	3	3	Right
2	4	2	1	2	3	Wrong
2	5	2.23	2	3	3	Wrong
3	1	2	1	2	3	Wrong
3	2	2.82	2	3	3	Wrong
3	4	2	2	3	3	Right
3	5	2	1	2	3	Wrong
3	6	2.82	3	2	3	Right
4	1	2.82	2	3	3	Wrong
4	2	2	1	3	3	Wrong
4	3	2	2	3	3	Right
4	5	2.82	2	2	3	Wrong
5	1	4	1	3	3	Wrong
5	2	2.23	2	3	3	Wrong
5	3	2	1	2	3	Wrong
5	4	2.82	2	3	3	Wrong
5	6	2	3	2	3	Wrong
6	3	2.82	1	3	3	Wrong
6	5	2	1	2	3	Wrong

Table 6. Implementation results for random topology for N = 6 nodes.

Node	Neighbors	Distance (m)	Implementation result	Real class	Communication range (R)	Result
1	2	2	2	2	3	Right
1	3	2.06	1	3	3	Wrong
1	4	1.41	2	2	3	Right
1	5	1.11	1	2	3	Wrong
2	1	2	1	2	3	Wrong
2	3	0.5	3	1	3	Wrong
2	4	1.41	1	2	3	Wrong
2	5	2.23	2	3	3	Wrong
3	1	2.06	1	3	3	Wrong
3	2	0.5	2	1	3	Wrong
3	4	1.11	2	2	3	Right
3	5	3	1	3	3	Wrong
3	6	1.11	3	2	3	Wrong
4	1	1.41	2	2	3	Right
4	2	1.41	1	2	3	Wrong
4	3	1.11	2	2	3	Right
4	5	1.41	2	2	3	Right
5	1	1.11	1	2	3	Wrong
5	2	2.23	2	3	3	Wrong
5	3	3	1	3	3	Wrong
5	4	1.41	2	2	3	Right
5	6	1.5	3	2	3	Wrong
6	3	1.11	1	2	3	Wrong
6	5	1.5	1	2	3	Wrong

5.4. Discussion

This work is directed as an empirical study of simulators and testbeds in order to allow a better choice in an evolution of a new protocol. It necessarily involves the study of existing simulators and testbeds in the literature, which enables the most used tools of the research community to select the best performers.

Simulation is a perfect environment, but the virtual world of simulator does not completely fit the characteristics of the real world. In fact, the Telosb and Micaz sensors have an omnidirectional antenna by definition. That is, the distance of communication covered by the sensor's transmission row is the same in all directions. Nevertheless, tests have shown that the distance covered (for a given transmission range) is not the same for the four sensor directions. For example: for node 3, the transmission range on one side of the Telosb sensor is 4 meters (m), for another side is 4m30. In addition, Micaz sensors do not work well if placed directly on the ground. It is therefore necessary to find objects like chairs or stones to place them on. Weather conditions have a major impact on communication, such as heat. In fact, the sensors run their program rapidly and efficiently in an indoor environment or in the shade. The number of runs (boot), that we can perform, was limited to 10,000 tries, and the energy of a sensor was limited. During the implementation process with testbeds, we faced some unavoidable situations, which greatly influenced the results of the tests carried out. All these remarks should be taken into consideration to infer results that are more correct.

6. CONCLUSION

Wireless sensor networks are widely used in various applications and domains. In this work, we have carried out a detailed experimentation study about implementation technics in WSNs. WSN testbed is designed to support experiments research in the real environments. Currently, experiments can be repeated to give more precise analysis results. In addition, simulation is an important approach for the implementation process, which is very useful for researchers. Although it cannot absolutely replace real experiments with testbeds. However, different communities of researchers can use a standard simulation framework, to increase the reliability and approval of the simulation results.

It is deeply important to offer researchers with this tools, to facilitate design stains, implementation and performance studies. Our study made practical remarks about each approach, because, if simulation gives enthusiastic results, the testbed can give results that are more faithful. It will give options to the researchers in their performances studies. The choice between testbeds and simulators should be based on the specific application as every WSN application has different requirements. These results urge us to develop more tools that include applications requirements in terms of mobility and heterogeneity of technologies, for example.

REFERENCES

- [1] D. Rong, (2016) "Wireless Sensor Networks in Smart Cities: The Monitoring of Water Distribution Networks Case", (Doctoral dissertation, KTH Royal Institute of Technology).
- [2] S. Kurkowski, T. Camp, & M. Colagrosso, (2005) "MANET simulation studies: the incredibles", *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(4), 50-61.
- [3] J. YICK, B. MUKHERJEE, D. GHOSAL, (2008)"Wireless sensor network survey", *Computer networks*, vol. 52, no 12, p. 2292-2330.
- [4] B. I. Bakare, and J. D. Enoch, (2019) "A Review of Simulation Techniques for Some Wireless Communication System", *International Journal of Electronics Communication and Computer Engineering*, 10(2).
- [5] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, (2014) "Wireless sensor networks: a survey on recent developments and potential synergies", *The Journal of supercomputing*, 68(1), 1-48.
- [6] A. S. Tonneau, N. Mitton , and J. Vandaele, (2015) "How to choose an experimentation platform for wireless sensor networks? A survey on static and mobile wireless sensor network experimentation facilities", *Ad Hoc Networks*, 30, 115-127.
- [7] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Marino, and J. García-Haro (2005, July) "Simulation tools for wireless sensor networks", In proceedings of the international symposium on performance evaluation of computer and telecommunication systems (SPECTS05) (p. 24).
- [8] S. A. B. Mussa, M. Manaf, K. Z. Ghafoor, and Z. Doukha, (2015, October) "Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives", In 2015 International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-8). IEEE.
- [9] P. Hurni, M. Anwander, G. Wagenknecht, T. Staub, and T. Braun, (2011, October) "Tarwis: A testbed management architecture for wireless sensor network testbeds", In Proceedings of the 7th International Conference on Network and Services Management (pp. 320-323). International Federation for Information Processing.
- [10] M. Pushpalatha, R. Venkataraman, K. Sornalakshmi, and T. Ramarao, (2013) "Implementation of Wireless Sensor Network Testbed-SRMSenseNet", *International Journal of Computer Applications*, 63(17).
- [11] E. Ertin, A. Arora, R. Ramnath, V. Naik, S. Bapat, V. Kulathumani, and M. Nesterenko, (2006, April) "Kansei: a testbed for sensing at scale", In Proceedings of the 5th international conference on Information processing in sensor networks (pp. 399-406). ACM.
- [12] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, and M. Singh, (2005, March) "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols", In *IEEE Wireless Communications and Networking Conference*, 2005 (Vol. 3, pp. 1664-1669). IEEE.

- [13] H. Hellbrück, M. Pagel, Köller, A., Bimschas, D., Pfisterer, D., and Fischer, S. (2011, June) “Using and operating wireless sensor network testbeds with WISEBED”, In 2011 The 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop (pp. 171-178). IEEE.
- [14] N. BOILLOT, D. DHOUTAUT, J. BOURGEOIS, (2013) “Efficient simulation environment of wireless radio communications in mems modular robots”, In : 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. IEEE, p. 638-645.
- [15] A. K. Dwivedi, and O. P. Vyas, (2011) “An exploratory study of experimental tools for wireless sensor networks”, *Wireless Sensor Network*, 3(07), 215.
- [16] I. T. DOWNARD, (2004) “Simulating sensor networks in ns-2”, NAVAL RESEARCH LAB WASHINGTON DC.
- [17] S. Keshav, (1988) “REAL: A network simulator”, Berkeley, Calif, USA: University of California.
- [18] Ns-3 Tutorial. <https://www.nsnam.org/docs/release/3.30/tutorial/singlehtml/index.html#futures>. [Accessed: 2019-09-25].
- [19] P. S. Katkar, and D. V. R. Ghorpade, (2016) “Comparative study of network simulator: NS2 and NS3”, *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(3).
- [20] P. LEVIS, N. LEE, M. WELSH, C. David, (2003) “TOSSIM: Accurate and scalable simulation of entire TinyOS applications”, In : Proceedings of the 1st international conference on Embedded networked sensor systems. ACM, p. 126-137.
- [21] E. PERLA, A. Ó. CATHÁIN, R. S. CARBAJO, M. Huggard, C. Mc Goldrick, (2008, October) “PowerTOSSIM z: realistic energy modelling for wireless sensor network environments”, In Proceedings of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (pp. 35-42). ACM.
- [22] B. L. TITZER, D. K. LEE, J. PALSBERG, “Avrora: Scalable sensor network simulation with precise timing”, In : IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, IEEE, p. 477-482, 2005.
- [23] UCLA Compilers Group. The AVR Simulation And Analysis Framework. <http://compilers.cs.ucla.edu/avrora/>, [Accessed: 2019-09-23].
- [24] F. Yu, and R. Jain, (2011) “A survey of wireless sensor network simulation tools”, Washington University in St. Louis, Department of Science and Engineering.
- [25] N. Ababneh, (2009, March) “Radio irregularity problem in wireless sensor networks: New experimental results”, In 2009 IEEE Sarnoff Symposium (pp. 1-5). IEEE.
- [26] C. H. Wu, and Y. C. Chung, (2007, May) “Heterogeneous wireless sensor network deployment and topology control based on irregular sensor model”, In International Conference on Grid and Pervasive Computing (pp. 78-88). Springer, Berlin, Heidelberg.
- [27] “Cognitive Radio Experimentation World”. Internet: <http://www.crew-project.eu/wilabt.html>, [Jan. 22, 2019].
- [28] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, (2010, May) “The w-iLab.t testbed”, In International Conference on Testbeds and Research Infrastructures (pp. 145-154). Springer, Berlin, Heidelberg.
- [29] “USC Networked Systems Laboratory”. Internet: <https://nsl.cs.usc.edu/Projects/Tutornet>, [Feb. 11, 2018].
- [30] G. WERNER-ALLEN, P. SWIESKOWSKI, M. WELSH, (2005) “Motelab: A wireless sensor network testbed”, In : Proceedings of the 4th international symposium on Information processing in sensor networks. IEEE Press, p. 68.
- [31] I. CHATZIGIANNAKIS, S. FISCHER, C. C. KONINIS, G. Mylonas, D. Pfisterer, (2009) “WISEBED: an open large-scale wireless sensor network testbed”, In : International Conference on Sensor Applications, Experimentation and Logistics. Springer, Berlin, Heidelberg, p. 68-87.
- [32] “FIT FUTURE INTERNET TESTING FACILITY”. Internet: <https://fit-equipex.fr/testbeds>, [Jan. 5, 2019].
- [33] “NITOS Wireless Sensor Platform”. Internet: <https://nitlab.inf.uth.gr/NITlab/hardware/sensors/wireless-sensor-platform>, [Jan. 28, 2019].
- [34] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, and L. Tassioulas, (2015, April) “Enabling open access to LTE network components; the NITOS testbed paradigm”, In Proceedings of the 2015 first IEEE Conference on Network Softwarization (NetSoft) (pp. 1-6). IEEE.

- [35] “An Open Platform for Developing, Deploying, and Accessing Planetary-Scale services”. Internet: <https://www.planet-lab.org/>, [Jan. 14, 2019].
- [36] B. CHUN, D. CULLER, T. ROSCOE, A. Bavier, L. Peterson, M. Wawrzoniak, M. Bowman, (2003) “Planetlab: an overlay testbed for broad-coverage services”, ACM SIGCOMM Computer Communication Review, vol. 33, no 3, p. 3-12.
- [37] M. OTT , I. SESKAR, R. SIRACCUSA, M. Singh, (2005) “Orbit testbed software architecture: Supporting experiments as a service”, In : First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities, IEEE, p. 136-145.
- [38] “Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT)”. Internet: <https://www.orbit-lab.org/>, [Jan. 26, 2019].
- [39] V. Handziski, A. K`opke, A. Willig, and A. Wolisz, (November 2005) “Twist : A scalable and reconfigurable wireless sensor network testbed for indoor deployments”, Technical Report TKN-05-008, Telecommunication Networks Group, Technische Universitat Berlin.
- [40] “emulab”. Internet: <https://www.emulab.net/portal/frontpage.php>, [Jan. 02, 2019].
- [41] K. Heurtefeux, and F. Valois, (2008, September) “Distributed qualitative localization for wireless sensor networks”, In International Conference on Ad-Hoc Networks and Wireless (pp. 218-229). Springer, Berlin, Heidelberg.
- [42] E. Niewiadomska-Szynkiewicz, (2012) “Localization in wireless sensor networks: Classification and evaluation of techniques”, International Journal of Applied Mathematics and Computer Science, 22(2), 281-297.
- [43] “Built With Emulab”. Internet: <http://www.emulab.arn.dz/index.php/en/pre>, [Feb. 02, 2019].

AUTHORS

Souhila Silmi is currently pursuing her Ph.D. thesis research at the computer science department of USTHB University of Science and Technology Houari Boumedienne, Algiers. She is a member of RIIMA laboratory at USTHB. She has research and teaching experiences. Her research interests include Ad hoc network, wireless sensor networks and Internet of Things.

Zouina Doukha received a PHD degree from USTHB University of Science and Technology Houari Boumedienne in 2016. She is associate professor and a member of RIIMA laboratory at USTHB. Her research activities include communication in VANETs, routing, localization and data collection in VSN, WSN and heterogeneous systems. She is also interested in virtualization, SDN and smart cities. She is serving as a reviewer for prestigious international conferences and journals.

Rebiha Kemcha received her MSc degree on Networks and Distributed Systems from University of Bejaia, Algeria in 2012. She worked on interconnection networks, their topologies and routing. She is PhD candidate at University of Boumerdes and teaching assistant in security of wireless networks. She works on evolutionary circuit design and design space exploration using multi-objective optimization algorithms.

Samira Moussaoui is a full Professor at the computer science department of the University of Sciences and Technologies Houari Boumediene (USTHB), Algiers. She received a PhD degree from USTHB University in 2007. She is Lecturer Researcher in the Department of computer science of USTHB, since 1988. Her research interests include mobile computing systems, mobile and wireless networks, and peer-to-peer systems.

AN ENHANCED LUCENE BASED SYSTEM FOR EFFICIENT DOCUMENT/INFORMATION RETRIEVAL

Alaidine Ben Ayed^{1,3}, Ismaïl Biskri^{2,3} and Jean-Guy Meunier³

¹Department of Computer Science,
Université du Québec à Montréal (UQAM), Canada
²Department of Mathematics and Computer Science,
Université du Québec à Trois-Rivières (UQTR), Canada
³LANCI : Laboratoire d'ANalyse Cognitive de l'Information,
Université du Québec à Montréal (UQAM), Canada

ABSTRACT

In this paper we implement a document retrieval system using the Lucene tool and we conduct some experiments in order to compare the efficiency of two different weighting schema: the well-known TF-IDF and the BM25. Then, we expand queries using a comparable corpus (wikipedia) and word embeddings. Obtained results show that the latter method (word embeddings) is a good way to achieve higher precision rates and retrieve more accurate documents.

KEYWORDS

Internet and Web Applications, Data and knowledge Representation, Document Retrieval.

1. INTRODUCTION

Document Retrieval (*DR*) is the process by which a collection of data is represented, stored, and searched for the purpose of knowledge discovery as a response to a user request (query) [1]. Note that with the advent of technology, it became possible to store huge amounts of data. So, the challenge has been always to find out useful document retrieval systems to be used on an everyday basis by a wide variety of users. Thus, *DR* -as a subfield of computer science- has become an important research area. IT is generally concerned by designing different indexing methods and searching techniques. Implementing an *DR* system involves a two-stage process: First, data is represented in a summarized format. This is known as the indexing process. Once, all the data is indexed, users can query the system in order to retrieve relevant information. The first stage takes place off-line. The end user is not directly involved in. The second stage includes filtering, searching, matching and ranking operations.

Most of the proposed *DR* systems are based on the cluster hypothesis [2]. Highly ranked documents relative to a given user query form a cluster that is easy to identify in the case of one simple query. For many complex queries there are query-specific clusters that contain many relevant documents [3] [4]. If those documents are not presented as the top of the result list, this would decrease the retrieval performance. There have been many attempts to propose ranking query-specific clusters techniques [5] [6] [7]. Most of proposed approaches simply compare

the representation of the cluster and the representation of the query. Some *DR* systems make use of additional features such as inter-cluster and cluster-document similarities [8] [9][10]. Also, Query expansion [11] is another way to boost *DR* performance.

Query expansion (*QE*) [11] has been a research field since the early 1960. [12] used *QE* as a technique for literature indexing and searching. [13] incorporated user's feedback to expand the query in order to improve the result of the retrieval process. [14] [15] proposed a collection-based term co-occurrence query expansion technique, while [16] [17] proposed a cluster-based one. Most of those techniques were tested on a small corpus with short queries and satisfactory results were obtained. Search engines were introduced in the 1990s. Previously proposed techniques were tested on bigger corpora sizes. We noticed that there was a loss in precision [18] [19]. Therefore, *QE* is still a hot search topic, especially in a context of big data.

To measure the accuracy of a *DR* system, there are generally two basic measures [20]: 1) *Precision*: the percentage of relevant retrieved documents and 2) *Recall*: the percentage of documents that are relevant to the query and were in fact retrieved. There is also a standard tool known as the TRECEVAL tool. It is commonly used by the TREC community for evaluating an ad hoc retrieval run, given the results file and a standard set of judged results.

In this paper we implement a document retrieval system using the Lucene toolkit [21]. Then we investigate the relevance of query expansion using parallel corpora and word embeddings to boost document retrieval precision. The next section describes the proposed system and gives details about the expansion process. The third one describes, and analyses obtained results. The last section concludes this paper and describes the future work.

2. METHODOLOGY

In this section, we present the structure of our Lucene system. First, we describe its core functions. In addition to that we describe some pre-processing operations as well as the evaluation process.

2.1. System Overview

Lucene is a powerful and scalable open source Java-based Search library. It can be easily integrated in any kind of application to add amazing search capabilities to it. It is generally used to index and search any kind of data whether it is structured or not. It provides the core operations for indexing and document searching.

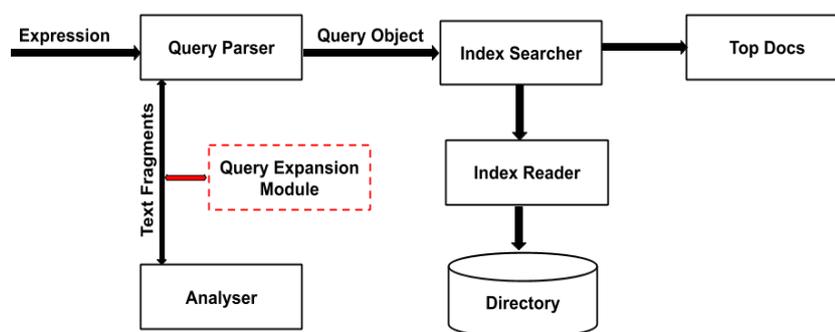


Figure 1: Proposed document retrieval system architecture.

Generally, a Search engine performs all or a few of the following operations illustrated by the above Figure. Implementing it requires performing the following actions:

- Acquire Raw Content: it is the first step. It consists in collecting the target contents used later to be queried in order to retrieve accurate documents.
- Building and analyzing the document: It consists simply in converting raw data to a given format that can be easily understood and interpreted.
- Indexing the document: The goal here is to index documents. So, the retrieval process will be based on certain keys instead of the entire content of the document.

The above operations are performed in an off-line mode. Once all the documents are indexed, users can conduct queries and retrieve documents using the above described system. In this case, an object query is instantiated using a bag of words present in the searched text. Then, the index database is checked to get the relevant details. Returned references are shown to the user. Note that different weighting schemes can be used in order to index documents. The most used ones are *tfidf* (the reference of the vectoral model) and *BM25* (the reference of the probabilistic model).

Typically, the *tf-idf* [22] [23] weight is composed by two terms: the first one measures how frequently a term occurs in a document. It computes the normalized Term Frequency (*TF*) which is the ratio of the number of times a word appears in a document by the total number of words in that document. the second term known as the inverse document frequency (*IDF*) measures how important a term is. It computes the ratio of the logarithm of the number of the documents by the number of documents where the specific term appears. *BM25* [24] ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity). It is generally defined as follows:

Given a query Q , containing keywords q_1, \dots, q_n the *BM25* score of a document D is:

$$score(D, Q) = \sum_{i=1}^n (q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}$$

where $f(q_i, D)$ is q_i 's term frequency in the document D , D is the length of the document D in words, and $avgdl$ is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]$ and $b = 0.75$. $IDF(q_i)$ is the *IDF* (inverse document frequency) weight of the query term q_i . It is usually computed as:

$$IDF(q_i) = \text{Log} \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i .

2.2. Query Expansion using a Comparable Corpora and Word Embeddings

In order to improve system accuracy, we proposed two techniques of query expansion. The first one uses Wikipedia as comparable corpus. The second one uses word embeddings. The main purpose is to make the query more informative while reserving its integrity.

2.2.1. Query Expansion using a Comparable Corpus

First, we use Wikipedia as a comparable corpus to expand short queries. For this purpose, we tested two slightly different approaches.

- *Query expansion by summary*: We extracted key-words from the query using the Rake algorithm [25]; a domain-independent method for automatically extracting keywords. we ranked keywords based on their order of importance, we took the most important one. Then, we used it to query Wikipedia. We summarized the first returned page; AKA, we made a short summary of one sentence and we concatenated it to the original query.
- *Query expansion by content*: We extracted key-words from the query using the Rake algorithm. we ranked keywords based on their order of importance. Then, we took the most important one and we used it to query Wikipedia. Therefore, we concatenated titles of the top returned pages to the original query.

2.2.2. Query Expansion using Word Embeddings

Word embeddings are also used to expand the queries. We assume that the concept expressed by a given word can be strengthened by adding to the query the bag of words that usually co-occur with it. For this purpose, we used the Gensim implementation of word2vec using three different models: glove-twitter-25, glove-twitter-200, fasttext-wiki-news-subwords-300 and glove-wiki-gigaword-300 [26].

3. EXPERIMENTS, RESULTS AND DISCUSSION

3.1. The Data Set

For experiments, we used a subset of the Trec dataset. It is a news corpus. It consists in a collection of 248500 journal article covering many domains such as economics, politics, science and technology, etc. First, we performed pre-processing of our corpus by removing stop words, applying stemming or lemmatization. Stemming is the process of transforming to the root word by removing common endings. Most common widely used stemming algorithms are Porter, Lancaster and Snowball. The latter one has been used in this project. In lemmatization, context and part of speech are used to determine the inflected form of the word and applies different rules for each part of speech to get the root word (lemma). Obtained results using different pre-processing strategies are reported in the next section.

The most frequently and important basic measures for document retrieval effectiveness are precision and recall [27]. Precision is simply the probability given that an item is retrieved it will be relevant and recall is the probability given that an item is relevant it will be retrieved. In this work, we use the TRECEVAL program [28] to evaluate the proposed system. It uses the mentioned above NIST evaluation procedures.

3.2. Results and Discussion

During our experiments we used three measures of precision:

- P5: Precision after 5 docs retrieved.
- P100: Precision after 1000 docs retrieved.
- MAP: Mean Average Precision

Obtained results are reported in Table 1, Table 2, Table 3 and Table 4.

Table 1: The importance of pre-processing

Data Type	original data			stemmed data		
Metric	P5	P10	Map	P5	P10	Map
Short queries	0.192	0.026	0.115	0.196	0.030	0.148
Long queries	0.194	0.030	0.139	0.236	0.037	0.148

Table 2: TFIDF VS. BM25

weighting schema	TFIDF			BM25		
Metric	P5	P10	Map	P5	P10	Map
Short queries	0.196	0.172	0.148	0.211	0.180	0.152
Long queries	0.236	0.266	0.148	0.242	0.221	0.161

Table 1 shows the system accuracy when using non-processed Vs. pre-processed data. It proves that pre-processing helps to achieve better precision rates. While, Table 2 shows results when using different weighting schema: better results are obtained by using the BM25 weighting schema. Notice here that we performed the same pre-processing before conducting experiences using different weighting schema.

Table 3: Obtained results when expanding queries by summary and content

Expansion Strategy	<i>O</i>			<i>T</i>			<i>S</i>		
Metric	P5	P10	Map	P5	P10	Map	P5	P10	map
Short queries	0.196	0.172	0.148	0.195	0.156	0.149	0.072	0.109	0.057

Table 3 displays obtained results when applying query expansion using a comparable corpus. We conducted three experiences: T (using expanded queries by title), S (expanded queries by topic) and we compared their results to those obtained by our original set of short queries. Notice here that we performed the same pre-processing and we used the same weighting schema. Obtained results show that expanding queries through titles of the top returned Wikipedia pages gives approximately the same precision rates with the original procedure. Whereas, expanding queries through the summary of the Wikipedia top page messes up the system accuracy.

Table 4: Obtained results when expanding queries through word embeddings

Model	<i>O</i>			<i>WE1</i>			<i>WE2</i>			<i>WE3</i>			<i>WE4</i>		
Metric	P5	P1000	Map	P5	P1000	Map	P5	P1000	map	P5	P1000	Map	P5	P1000	map
Short queries	0.196	0.030	0.148	0.164	0.026	0.018	0.01188	0.027	0.116	0.204	0.032	0.125	0.216	0.028	0.132

Table 4 shows results when expanding queries using word embeddings. we used the Gensim implementation of word2vec. We tested three different models: glove-twitter-25 (*WE1*), glove-twitter-200 (*WE2*), fasttext-wiki-news-subwords-300 (*WE3*) and glove-wiki-gigaword-300 (*WE4*). Obtained results show that the system accuracy can be enhanced when taking in consideration the top 5 returned results. To achieve this goal, we should use the appropriate model: *WE3*, which is trained with a news corpus or *WE4* which is trained using a very huge amount of data.

4. CONCLUSIONS AND FUTURE WORK

In this work, a *DR* system based on the Lucene toolkit is presented. Different weighting schema are tested. Results show that the probabilistic model (*BM25*) performs the vectoral one (*TFIDF*). Also, lead experiments show that query expansion using word embeddings improves the overall system precision. Meanwhile, using a comparable corpus doesn't necessarily lead to the same result. This paper can be improved by:

- Testing an interactive query expansion technique: experimental results show that the query expansion using a comparable corpus does not lead to higher precision rates. The precision rate depends on the efficiency of the Rake key word extractor algorithm. The main idea is to let users validate the automatically extracted keywords used later during the query expansion process.
- Testing a hybrid technique of query expansion: word embeddings can be applied on the result of the interactive query expansion phase. This may boost the system performance since the interactive query expansion will guarantee the use of significant words of the query. Also, using word embeddings will ensure retrieving relevant documents which do not necessarily contain words used in the query.

ACKNOWLEDGEMENTS

The authors would like to thank Natural Sciences and Engineering Research Council of Canada for financing this work.

REFERENCES

- [1] Anwar A. Alhenshiri, Web Information Retrieval and Search Engines Techniques, 2010, Al-Satil journal, PP:55-92
- [2] Fiana R, Oren K., Ranking Document Clusters Using Markov Random Fields, Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013
- [3] Tombros A., Villa R., and van Rijsbergen C. The effectiveness of query-specific hierarchic clustering in information retrieval. *Process. Manage.*, 38(4):559–582, 2002.
- [4] Liu X. and Croft W. B. Experiments on retrieval of optimal clusters. Technical Report IR-478, Canter for Intelligent Information Retrieval (CIIR), University of Massachusetts, 2006
- [5] Leuski A. Evaluating document clustering for interactive information retrieval. In *Proc. of CIKM*, pages 33–40, 2001.
- [6] Liu X. and Croft W. B. Cluster-based retrieval using language models. In *Proc. of SIGIR*, pages 186–193, 2004
- [7] Liu X. and Croft W. B., Evaluating text representations for retrieval of the best group of documents. In *Proc. of ECIR*, pages 454–462, 2008
- [8] Kurland O. and Lee L. Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proc. Of SIGIR*, pages 83–90, 2006.
- [9] Kurland O. and Domshlak C., A rank-aggregation approach to searching for optimal query-specific clusters. In *Proc. of SIGIR*, pages 547–554, 2008
- [10] Kurland O. and Krikon E., The opposite of smoothing: A language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research (JAIR)*, 41:367–395, 2011.
- [11] H. K. Azad, A. D., Query Expansion Techniques for Information Retrieval: a Survey. 2017.
- [12] Maron, M.E., Kuhns, J.L.: On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)* 7(3), 216-244, 1960.
- [13] Rocchio, J.J.: Relevance feedback in information retrieval, 1971.
- [14] Jones, K.S.: Automatic keyword classification for information retrieval, 197.

- [15] van Rijsbergen, C.J.: A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of documentation* 33(2), 106-119, 1977.
- [16] Jardine, N., van Rijsbergen, C.J.: The use of hierarchic clustering in information retrieval. *Information storage and retrieval* 7(5), 217-240, 1971.
- [17] Minker, J., Wilson, G.A., Zimmerman, B.H.: An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval* 8(6), 329-348, 1972.
- [18] Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41, 288-297, 1990.
- [19] Harman, D.: *Relevance feedback and other query modification techniques*, 1992.
- [20] R. Sagayam, S. Srinivasan, S. Roshni, A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques, *IJCIER*, sep 2012, Vol. 2 Issue. 5, PP: 1443-1444.
- [21] <https://lucene.apache.org/core/>
- [22] Breiting, C.; Gipp, B.; Langer, S. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*. 2015. 17 (4): 305-338.
- [23] Hiemstra, Djoerd. A probabilistic justification for using tfidf term weighting in information retrieval. *International Journal on Digital Libraries* 3.2 (2000): 131-139.
- [24] Stephen E. R.; Steve W.; Susan J.; Micheline H-B. & Mike G. Okapi at TREC3. *Proceedings of the Third Text Retrieval Conference (TREC 1994)*. Gaithersburg, USA.
- [25] Stuart J-R, Wendy E-C. Vernon L-C. And Nicholas O-c, *Rapid Automatic Keyword Extraction for Information Retrieval and Analysis*, G06F17/30616 Selection or weighting of terms for indexing, USA, 2009
- [26] Jeffrey P., Richard S., Christopher D-M, *GloVe: Global Vectors for Word Representation*.
- [27] David M.W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies*. 2 (1): 3763.
- [28] <https://trec.nist.gov/>

AUTHORS

Alaidine Ben Ayed is a PhD. candidate in cognitive computer science at Université du Québec à Montréal (UQAM), Canada. His research focuses on artificial intelligence, natural language processing (Text summarization and conceptual analysis) and information retrieval.



Ismail Biskri is a professor in the Department of Mathematics and Computer Science at the Université du Québec à Trois-Rivières (UQTR), Canada. His research focuses mainly on artificial intelligence, computational linguistics, combinatory logic, natural language processing and information retrieval



Jean Guy Meunier PhD. is a research professor at UQAM, co-director of the Cognitive Information Analysis Laboratory (LANCI), member of the Institute of Cognitive Sciences at UQAM and member of the Centre for Research in Digital Humanities (CRHN), full member of the International Academy of Philosophy of Science (Brussels).



MULTIPLE LAYERS OF FUZZY LOGIC TO QUANTIFY VULNERABILITIES IN IOT

Mohammad Shojaeshafiei¹, Letha Etzkorn¹ and Michael Anderson²

¹Department of Computer Science, The University of Alabama in Huntsville, Huntsville, USA

² Department of Civil and Environmental Engineering, The University of Alabama in Huntsville, Huntsville, USA

ABSTRACT

Quantifying vulnerabilities of network systems has been a highly controversial issue in the fields of network security and IoT. Much research has been conducted on this purpose; however, these have many ambiguities and uncertainties. In this paper, we investigate the quantification of vulnerability in the Department of Transportation (DOT) as our proof of concept. We initiate the analysis of security requirements, using Security Quality Requirements Engineering (SQUARE) for security requirements elicitation. Then we apply published security standards such as NIST SP-800 and ISO 27001 to map our security factors and sub-factors. Finally, we propose our Multi-layered Fuzzy Logic (MFL) approach based on Goal question Metrics (GQM) to quantify network security and IoT (Mobile Devices) vulnerability in DOT.

KEYWORDS

Computer Network, Network Security, Mobile Devices, Fuzzy Logic, Vulnerability, Cyber Security

1. INTRODUCTION

Today we see drastic development and improvement of Information and Communication Technology (ICT). It is inevitable that many changes in computer networks result in security complications. Thus, when we analyze an organization's network system we are required to consider its potential vulnerabilities for any unexpected attack or information leakage. many types of methodologies and procedures have been proposed using proper measurement approaches to tackle such vulnerabilities in an organization's network system [1][2][3]. Mainly, these methodologies are divided into two categories: Qualitative and quantitative [4]. In the past [5] Multi-layered Fuzzy Logic has been used to quantify previously qualitative concepts. In this paper, we propose a Multi-layered Fuzzy Logic (MFL) approach to quantify potential vulnerabilities of the Department of Transportation (DOT). It comprises the computer network's security and Mobile Devices as regards to IoT of DOT. For that, we have considered all aspects of DOT's computer networks and Mobile Devices for security analysis. With a thorough security analysis to measure vulnerabilities, we have listed all security aspects of computer network and Mobile Devices as the security factors in a top-down manner from major to minor. Afterward, we evaluated all security factors to check if they overlap (in order to remove redundancy). Finally, we accepted Availability, Integrity, Accuracy, and Confidentiality for computer network vulnerability analysis, and Enterprise Mobile Management (EMM), User Access Control(UAC), and Encryption for Mobile Devices for consideration.

We apply the Goal Question Metrics (GQM) [6] approach to provide the required input for the MFL approach. The input for MFL will be in the form of qualitative processes that are derived from the computer network and security expert evaluation in DOT based on standard security questions mapped to NIST SP800-53 and ISO 27001 standards. The entire procedure is addressed in section 3.

The rest of this paper is organized as follows: Section 2 presents the background of the work to address the Fuzzy Logic in related work, computer network vulnerability quantification and IoT's vulnerability measurement. Section 3 provides a description of our methodology in GQM, security standards, factors and sub-factors of network security and Mobile Devices. Section 4 addresses Multi-layered Fuzzy Logic (MFL) implementation.

2. BACKGROUND

2.1. Fuzzy Logic and Related Work

Vulnerability measurement processes almost always result in a high degree of uncertainty. Because whenever we discuss the security of a network, we describe it as a linguistic variable form of 'secure', or 'not secure', that causes imprecision and vagueness. Therefore, we are not able to evaluate the accuracy of evaluation based on linguistic variables. In 1965, Lotfi Zadeh [7] at the University of California at Berkley proposed Fuzzy logic (and proved it mathematically). This method says that conventional computer logic is not able to work on data manipulation when the data carries the vagueness of human linguistic propensity. One of the advantages of the Fuzzy Logic methodology for vulnerability measurement is that the implementations based on mathematical models are reliable in all aspects of security analysis based on previous analysis of indication motors [8].

It is very important to know the degree of truth in Fuzzy Logic. Membership Function (MF) [9] is a pivotal component in Fuzzy Logic to provide such a degree of truth. It defines a function that specifies the degree of "belongingness" of an input to a set. The value of MF is always limited in the interval of [0-1]. The most common forms of MF are Triangular, Trapezoidal, piecewise Linear, Gaussian, and Singleton [10]. Figure 1 depicts an example of Triangular MF for temperature in a sub-set of cold, cool, normal, warm and hot.

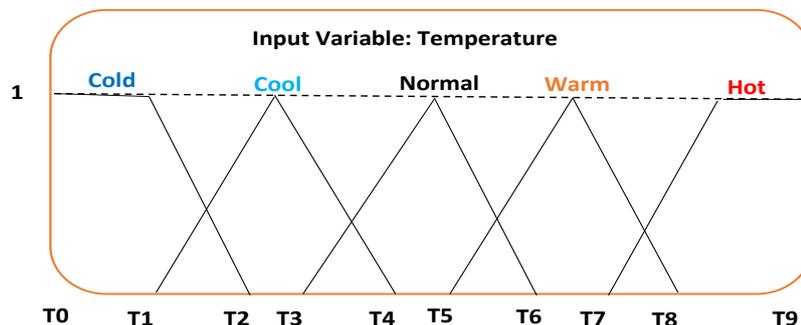


Figure 1. Triangular MF for Temperature

The triangular function is defined as follows:

“ For a fuzzy set, A on the universe of discourse X is defined as $\mu_A: X \rightarrow [0,1]$, where each element of X is in an area between 0 and 1 that quantifies the grade of membership of the element in X to

the fuzzy set A. It is defined by a lower bound a and an upper bound b and the value m where $a < m < b$ [7] (Table 1).

Table 1. Triangular Function Equation.

$\mu_A(x) =$	0	If: $x \leq a$
	$(x-a)/(m-a)$	If: $a < x \leq m$
	$(b-x)/(x-m)$	If: $m < x \leq b$
	0	If: $x \geq b$

A Fuzzy Logic system is a nonlinear mapping of the input data to a scalar output data and it has four major steps as shown in Figure 2.

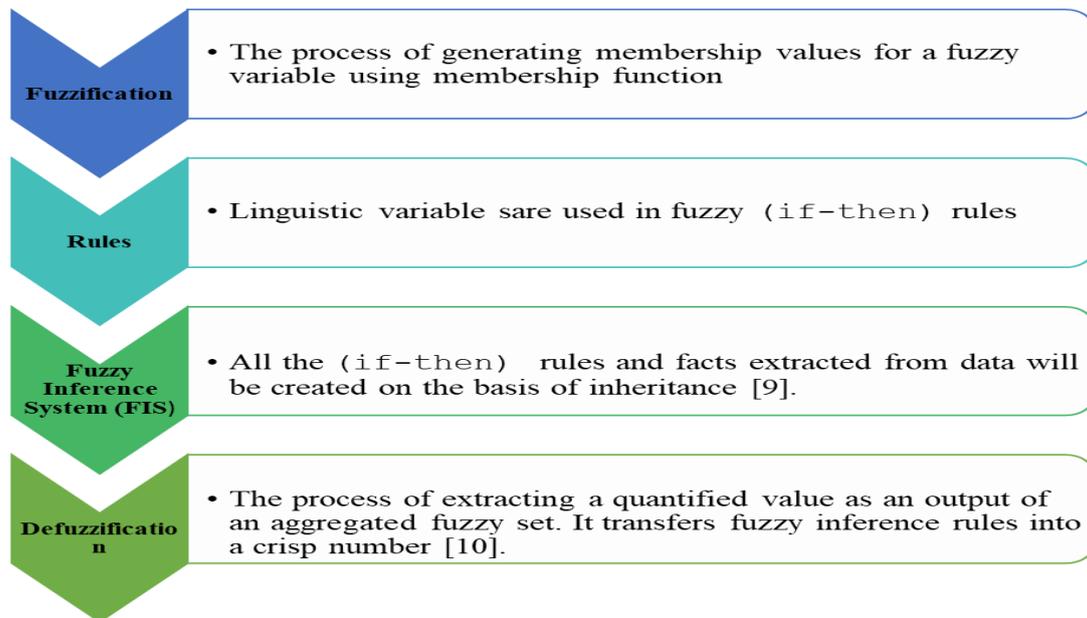


Figure 2. Fuzzy Logic Steps

Some authors have conducted research using the Mamdani model [13] of Fuzzy Inference to identify and measure potential threats and risks [14][15]. They applied this model to software to find out if the software modules are prone to attack or not. The main goal of their methodology was to eliminate training and testing phases while building up the procedure of software fault prediction using soft computing methods.

Kamongi et al. [16] proposed a method of ranking cloud system vulnerability and implemented it with vulnerability discovery on a cloud web application: the result was a list of ranked vulnerabilities associated with an attack path. They conceptualized cloud vulnerabilities as the attack paths from a pre-generated attack graph. Each path consists of information in regard to vulnerabilities of pre and post conditions that are required to be met for a successful security breach.

Anikin et al. [17] proposed a quantitative information security risk assessment in computer networks. They applied Fuzzy Logic and analytic hierarchy processes to evaluate the impact and quantify the value of a specific threat. They presented the vulnerability risk assessment based on Vulnerability Scoring System (CVSS) [18], but with the elimination of CVSS' barriers to risk quantification. They suggested fuzzy *if-then* rules for decision support systems with discrete output that was obtained from decision making in oil production.

Kuang et al. [19] conducted research on network security situation forecast using Fuzzy logic based on the Markov model. Their methodology is based on the combination of safe behavior historical data with the level of threats in the system. They analyzed branch prediction based on the Markov model and membership degree evaluation based on the Fuzzy system of information security vulnerabilities. Finally, they provided an integrated version of security situation prediction based on Fuzzy Logic. For verification, they used data from KDD CUP99's data as training data and the DARPA2000's data as testing data to obtain vulnerability information [20][21].

2.2. Internet of Thing (IoT) Vulnerability Assessment

Internet of Things (IoT) is defined as the structure and combination of multiple devices that are connected to each other through the internet [22]. Interaction with and between IoT devices has changed over time. These changes can be due to either software or hardware variations. Such changes, which result in a loss of integrity, are nowadays the major subject of security concerns in IoT. Lack of integrity in IoT causes a considerable cost for service providers and users because the adversary is able to take advantage of such security holes and cause irreparable harm. Millions of IP security cameras, doorbells, etc. are vulnerable to attack or can be hijacked to work against the devices' owners. In organizations' security policies, the inherent risk of Bring Your Own Device (BYOD) made the IT departments to become more restrictive and required extensive monitoring of portable devices. Mobile Device Management (MDM) tools are also applied as security software to monitor and manage employees' mobile devices. However, with all of these strict policies and monitoring, a wide variety of vulnerabilities still threatening organizations and put them into a serious risk of the data breach. Much research has been studied in this area as a major concern of today's internet security. However, the problems in this area are not yet solved.

Williams et al. [23] conducted research on vulnerability assessment of consumer IoT devices. They used Nessus [24] for vulnerability assessment since it has the capability of scanning many devices at a time. They scanned a large number of devices and extracted the result in a set such as 'critical', 'high', 'medium', and 'low' risks. They used Nessus for vulnerability assessment of IoT in different categories including devices of the home, workplace, and cities.

Patton et al. [25] studied the vulnerable devices on the IoT and provided an evaluation of existing vulnerabilities in the IoT system. They applied Shodan, a search engine for IoT, that uses its database to maintain past scans. In the first step, they built an executed version of python scripts to utilize the Shodan API to interface with Shodan. Then they parsed device headers into MySQL database and used a password database; with a python script, they tested the password against IoT devices to capture vulnerabilities. They used the list of IP addresses collected from Shodan to scan thousands of devices to check if the default login credentials work. The result of vulnerabilities varied from %0.44 to %40.

2.2.1. Mobile Devices

Based on Gartner Research's forecast [26] there will be 20.4 billion IoT devices by the end of 2020 and almost 75 billion devices in 2025. With the expansion of the network systems in organizations, increasing the number of IoT devices due to the immense availability of connectivity devices is inevitable. This expectation of exponential proliferation is reasonable because enterprises make a large profit through the application of these ubiquitous mobile devices.

On the other hand, IoT (including all mobile devices) put the enterprises at a high risk of being susceptible to attack through potential vulnerabilities. One of the most important parts of the cybersecurity domain is vulnerability measurements in Mobile Devices as a portable form of IoT in organizations such as DOT. There have been many arrangements for Mobile Devices policies and restrictions including building multiple layers of protection in agencies to keep their information safe from cyber-attacks. The more the employees in agencies use mobile devices the more the probability of cyber-attacks will emerge. The support layers to bolster the security of Mobile Devices consist of providing anti-malware software, secure mobile communications (using VPNs and requiring strong encryption and authentication), control of third-party's software, performing penetration testing to check for vulnerabilities, auditing, etc. Nevertheless, a clear-cut quantification procedure to measure vulnerabilities of Mobile Devices has always been neglected, or at least, it has not been taken as seriously as possible. Thus, in this paper, we consider all aspects of Mobile Devices as part of IoT in DOT to provide a quantifiable measurement methodology for the potential vulnerabilities of a system.

3. METHODOLOGY

3.1. Security Requirements

Prior to the vulnerability measurement process of network security and Mobile Devices, we have to assure that our network design and implementation comply with standard security requirements as a pre-requisite of network design. One of the most important security flaws in organizations that leads them to be prone to cybersecurity attacks is the lack of a comprehensive and precise engineered design of security requirements [27]. It is implausible to design security requirements for a network system that guarantee no flaws, but still, organizations must take this step as a fundamental stage of attack prevention. The detection and correction phase of security requirements development costs 10 to 200 times less than the detection of flaws after system deployment in the field [27]. For security requirements elicitation there have been several proposed models such as Multilateral Security Requirements Analysis (MSRA) and Goal Base Requirements Analysis Record (GBRAM). In this paper, we apply the Security Quality Requirements Engineering (SQUARE) [28] for security requirements elicitation because it considers risk assessment and quality assurance simultaneously to maintain availability, integrity, and confidentiality of the security requirement goals. SQUARE is a process model developed at Carnegie Mellon University to elicit, categorize and prioritize security requirements for IT departments and applications.

In order to generate final deliverable prioritized security requirements in the system, SQUARE provides nine fundamental steps for security requirements elicitation. We adopted this model and extracted several major points to be highlighted as fundamental security requirements in DOT. These steps are Integrity, Physical Security, Authentication, Access Control, Availability, Audition, and Authorization. Therefore, the very first step of vulnerability measurement in DOT

is following these security requirement elicitation processes to design all components of security in the system for network security and Mobile Devices (IoT).

3.2. Goal Question Metrics

Goal Question Metrics (GQM) is an approach for developing a model based on the goals of the project. This model was originally developed by Basili [29] at the University of Maryland to measure software quality. The metric measurement model is proposed in three steps: conceptual model (Goal), operational model (Question) and quantitative model (Metrics) [30]. One good aspect of this model's application is that there may be many goals and consequently multiple times questions for a project, but the rate of metrics does not grow at the same rate as goals or questions. These three steps are decomposed to six steps in detail for metric measurements in the software product as follows:

1. Develop project goals.
2. Generate questions that describe the goals in step 1.
3. Define the specific measures to answer the questions designed in step 2.
4. Data collection mechanisms.
5. Validate and analyze data that are obtained in step 4.
6. Data analysis to assess the conformance of the goals.

Based on the analysis and consideration of security requirements using SQUARE model for network security and Mobile Devices in DOT we design questions accordingly for each security factor in a top-down manner to reach the desired goal (vulnerability quantification). The main reason for using GQM in our research is that it is a de facto standard for quality metrics in software engineering and is therefore widely accepted and respected [29]. Our use of GQM is analogous to how it has been previously used in software metrics. Based on the security requirements of DOT we designed the goals (numerical vulnerability value of each security factor) that are traceable to a set of quantifiable questions [31]. In our previous research [32], we introduced a hierarchy model to identify the security requirements of organizations using GQM to be a primary work of vulnerability measurement that enable us to trace from security requirements to security metrics.

3.3. Security Standards

After identifying the security requirements and determining security goals and questions based on GQM to measure vulnerability, we map every possible security question derived from essential factors of network security and Mobile Devices to the currently published security standards from NIST SP800-53 and ISO 27001. Passing through this process is to make sure that we are following all the available security standards.

ISO/IEC 27001:2013 [33] is an international standard to specify an information security management system (ISMS). This standard helps organizations focus on three key aspects of information: confidentiality, integrity, and availability. It examines the organization's risk, threat and potential vulnerabilities, provides all information security controls in order to reinforce strategies for risk avoidance, and finally, it makes sure that all security management processes and security controls meet the organizations' security needs.

Another purpose of this standard is to verify the proper selection of proportionate security controls to provide an acceptable amount of protection for assets. In sum, its main features are information security policies, communication, and operational management, access control,

information system acquisition, organization of information security, asset management, business continuity management, human resources security and physical security [34].

NIST SP800-53 [35] is a publication of the National Institute of Standards and Technology (US) that specifies a set consisting of 198 security controls. Controls are categorized into three main groups: technical, operational and management. These three groups can characterize several subgroups such as access control, awareness and training, audit and accountability, security assessment and authorization, configuration management, contingency planning, identification and authorization, system and information integrity, etc.

3.4. Security Factors and Sub-factors in DOT

While analyzing the main components of network security and Mobile Devices (as part of IoT) in DOT, we designed each security component as security factors and sub-factors at each level. Table 2 and Table 3 describe each security factor and sub-factor based on the published security standards of NIST SP800-53 and ISO 27001.

Table 2. Network Security Factors and Sub-factors in DOT.

Network Security	Availability	Redundancy	N/A
		Monitoring & Alerting	
		Backup	
		Load Balancing(proxy)	
		Disaster Readiness	
		Disaster Recovery Plan	
		Quality Testing	
		Wireless Security	
	Accuracy	Data Accuracy	Annual Review
			VAM(Vulnerability Assessment & Management)
			Data labeling
	Integrity	Data security	Encrypted Communication
			Supply Chain
		Data Quality and Integrity	N/A
		Software Security	Software Update
			Outsource Development
			Threat Detection Update
		Quality Testing	N/A
	Wireless Security	N/A	
	Confidentiality	Authorization and Identification	Username and Password
			Program Access
Authentication		Data Classification,	
		Two-factor Authentication	
		Use Geographic Location as an Authentication Factor	
User Access Control (UAC)		N/A	
Encryption		N/A	
Quality Testing	N/A		

Table 3. Mobile Devices' Security Factors and Sub-factors in DOT.

Mobile Devices	Enterprise Mobility Management (EMM)	Mobile Device Management System (MDM)	Lost Devices	Security Questions or Challenges
				Lost or Stolen Reports
			Inventory of Mobile Devices	
			Automatic Lockout Screen	
			Jailbreaking Prevention	
		Mobile Application Management System (MAM)	Operating System Management Program	
	User Access Control (UAC)	Remote Data Wipe	MDM	
			EMM	
		Username and Password	Credential Prompt	
			Log Viewer	
		External Memory Policy	N/A	
		Bring Your Own Device (BYOD)	N/A	
		Monitoring Policy	N/A	
	Encryption	Data and Device Encryption	Encryption Algorithms	

Since our goal in this research is to quantify vulnerabilities in DOT from the perspective of network security and IOT, we designed relevant security questions to map all security factors and sub-factors mentioned in Table 2 and Table 3 based on the GQM model. The questionnaire is designed in a top-down manner to assign one or more question(s) for vulnerability measurement depending on the type of questions. The questionnaire will be answered by the computer network and security experts of DOT. A few numbers of questions are chosen from the questionnaire and are listed in Table 4.

Table 4. Security Questions derived from GQM mapped to Security Standards

Security Factor	Question
Availability	Does DOT make sure security mechanisms and redundancies are implemented to protect equipment from utility service outages (e.g., power failures, network disruptions, etc.)
Integrity	How often does DOT ensure that data does not migrate beyond a defined geographical residency?
Accuracy	How often does your organization consider annual review including third party providers upon which their information supply chain depends?
Confidentiality	Does your agency require two-factor authentication for remote access? (e.g. token is used in addition to a username, and password).

Enterprise Mobility Management	How often does your agency require/remind employees to report their mobile devices' lost or stolen?
User Access Control (UAC)	Does your organization require and enforce via technical controls an automatic lockout screen for mobile devices or any company-owned devices?
Encryption	How often does your organization's mobile device policy require the use of encryption for either the entire device or for data identified as sensitive enforceable through technology controls for all mobile devices?

3.5. Multi-Layered Fuzzy Logic

The main reason for using Fuzzy Logic is that it is one of the most reliable mathematical tools to model problems that have the most inaccuracy and uncertainty [7]. Another crucial reason for using Fuzzy Logic in this research is that we are dealing with linguistic variables to qualitatively determine the value of the security as 'good' or 'bad'. As we described the Triangular Fuzzy model MF in section 2, we will apply it for vulnerability quantification because it provides a simple Fuzzy Inference System (FIS) that correlates the vulnerability attributes quantitatively in fuzzification processes. We use the most common properties of fuzzy logic for A_{fuzzy} and B_{fuzzy} as follows:

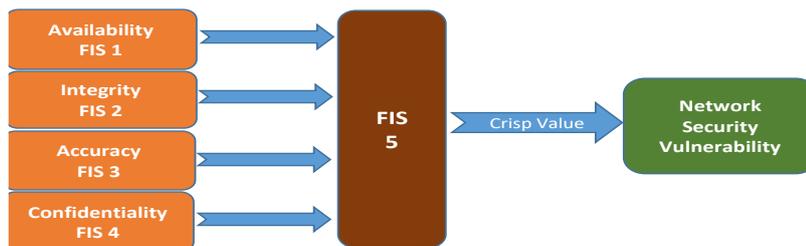
$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \mid x \in X, \quad \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \mid x \in X$$

In the Defuzzification process, this model generates a crisp number derived from the fuzzy set. There are several Defuzzification methods such as Center of Area (COA), Bisector of Area (BOA), largest of Maximum (LOM), Mean of Maximum (MOM), Smallest of Maximum, etc [36]. We use the COA method [37] since this method considers, determines the center of the area of fuzzy set and returns the corresponding crisp value of that [38][12].

To measure the vulnerability of the network system and Mobile Devices in DOT, we follow a few important steps from the analysis part in the cybersecurity domain to the final step of measurement. Figure 3 depicts the order of the processes.



Figure 3. vulnerability Measurement Steps in DOT



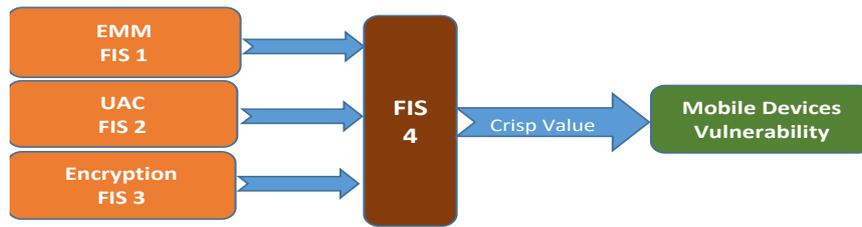
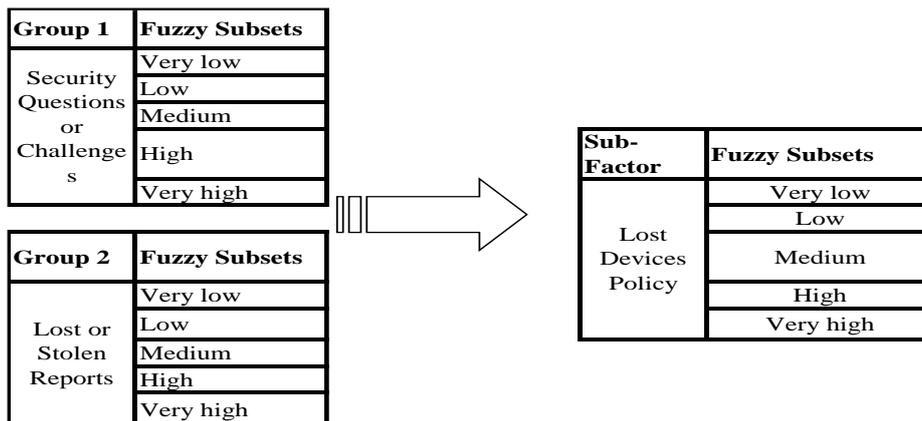


Figure 4. MFL for Network Security and Mobile Devices (IoT) in DOT

As it shows in Figure 4 we define a Fuzzy Inference System (FIS) process separately for each security factor in a backward flow from sub-score (leaves) to the main factors (nodes). In the next phase, the results of the first-step-FIS are being processed cumulatively in the second-step-FIS. The result is a crisp value to measure vulnerabilities of the factors based on the FIS rules that are defined for linguistic variables. The main steps to determine the parameters of FIS are the MF calculation mechanism of linguistic values to maintain them in a database for both antecedent (*if*) and consequent (*then*), plus the Fuzzy reasoning mechanism from the number of used Fuzzy rules [39].

When security experts in the DOT answer the questionnaire for each sub-factor, the answers will be categorized to appropriate security Groups based on the role of each sub-factor in network security, and Mobile Devices. The reason for this classification is that the answers are in the format of fuzzy subsets such as ‘very low’, ‘low’, ‘medium’, ‘high’ and ‘very high’. The weights from 0 to 10 are assigned to each sub-factor in the questionnaire to determine what fuzzy subset each sub-factor belongs to. In this case, the factor Enterprise Mobility Management (EMM) is the result of the combination of all sub-factors in Group 1 (Mobile Device Management System (MDM)) and Group 2 (Mobile Application Management System (MAM)). To achieve MDM and MAM we apply FIS accordingly based on their own sub-groups derived from Table 3. As shown in Figure 5 the interior FIS layers of MDM are generated from security questions, lost or stolen reports, jailbreaking prevention, automatic lockout and inventory of mobile devices. After obtaining all required crisp values from MFL, we measure the final value of vulnerability from the aggregation of the previous values.



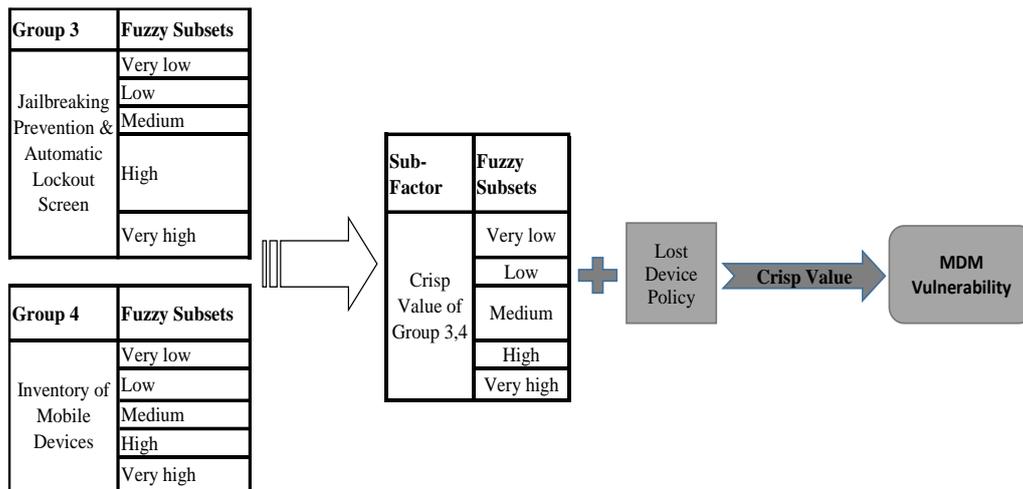


Figure 5. Fuzzy Subset implementation for EMM based on interior FIS.

In order to convert the expert answers of the questions from the form of linguistic values to the form of Fuzzy subset, we need to define the concepts of the values ‘very low’, ‘low’, ‘medium’, ‘high’ and ‘very high’ for each question. Since the implementation and the full discussion of all components of network security and Mobile Devices are beyond the scope of this paper we show only the implementation of Mobile Devices (IoT) in DOT. Table 5 addresses these definitions based on the answers to each question in Mobile Devices’ factors and sub-factors.

Table 5. Description of Weighted Questions to convert to Fuzzy Subsets

Sub-factors	Description in Fuzzy sets	
Security Questions	Very low	DOT never asks security questions or challenges to their employees to prevent unwanted access to Mobile Devices.
	Low	DOT sometimes asks security questions to their employees to prevent unwanted access to Mobile Devices.
	Medium	DOT regularly provides mechanisms to prevent unwanted access to Mobile Devices.
	High	DOT usually provides mechanisms to prevent unwanted access to Mobile Devices.
	Very High	DOT always asks extra security questions and challenges to their employees coupled with providing mechanisms to prevent unwanted access to Mobile Devices.
Lost or Stolen Reports	Very Low	DOT never requires their employees to report if their Mobile Devices are lost or stolen.
	Low	DOT rarely requires their employees to report if their Mobile Devices are lost or stolen.
	Medium	DOT regularly requires their employees to report if their Mobile Devices are lost or stolen.
	High	DOT usually requires their employees to report if their Mobile Devices are lost or stolen, if so, they usually have a security plan for data protection of such devices.
	Very High	DOT always requires their employees to report if their Mobile Devices are lost or stolen, if so, they always have a strong security plan for data protection of such devices.
Inventory of Mobile Devices	Very Low	DOT never maintains an inventory of all Mobile Devices.
	Low	DOT sometimes maintains an inventory of all Mobile Devices.
	Medium	DOT regularly maintains an inventory of all Mobile Devices.
	High	DOT usually maintains an inventory of all Mobile Devices storing and

		accessing company data which includes the status of OS, patch level, lost, decommissioned and device assignee.
	Very High	DOT always maintains an inventory of all Mobile Devices storing and accessing company data which includes the status of OS, patch level, lost, decommissioned and device assignee.
Automatic Lockout Screen	Very Low	DOT does not have any plan for Mobile Devices' automatic lockout screen.
	Low	It is not DOT's priority to have an automatic lockout screen plan for Mobile Devices.
	Medium	It is important for DOT to have an automatic Lockout Screen plan for Mobile Devices but they are not very strict on that.
	High	DOT's Mobile Devices policies usually prohibit the circumvention of built-in security controls on mobile devices such as jailbreaking or rooting but they are not very strict on that.
	Very High	DOT's Mobile Devices policies always prohibit the circumvention of built-in security controls on mobile devices such as jailbreaking or rooting.
MAM	Very low	DOT does not provide any Mobile Application Management (MAM) plan for Mobile Devices' security.
	Low	DOT sometimes performs a Mobile Application Management (MAM) plan for Mobile Devices' security.
	Medium	It is not a priority for DOT to perform Mobile Application Management (MAM) for mobile devices' security maintenance, but it is part of their security plan.
	High	DOT usually performs Mobile Application Management (MAM) to maintain the security of Mobile Devices.
	Very High	DOT always performs or implements Mobile Application Management (MAM) to maintain the security of Mobile Devices and manages all changes to the devices' OS.
Remote Data Wipe	Very Low	DOT's IT department never has a plan for remote data wipe for Mobile Devices.
	Low	DOT's IT department rarely provides remote data wipe or corporate data wipe for Mobile Devices.
	Medium	DOT's IT department regularly provides remote data wipe or corporate data wipe for Mobile Devices.
	High	DOT's IT department usually provides remote wipe or corporate data wipe for Mobile Devices.
	Very High	DOT's IT department always provides remote wipe or corporate data wipe for all company-accepted BYOD devices or any mobile devices
Monitoring	Very Low	DOT does not have any controls on unauthorized software installation.
	Low	DOT rarely controls unauthorized software installation.
	Medium	DOT has a regular plan for unauthorized software installation, but not strictly.
	High	DOT usually has controls in place to restrict and monitor the installation of unauthorized software onto the Mobile Devices.
	Very High	DOT always has controls in place to restrict and monitor the installation of unauthorized software onto the Mobile Devices.
User name and Password	Very Low	DOT never asks employees to change the username and password of their Mobile Devices.
	Low	DOT sometimes asks employees to change the username and password of their Mobile Devices but they are not required.
	Medium	DOT regularly requires employees to change the username and password of their Mobile Devices.
	High	DOT usually requires employees to change the username and password of their Mobile Devices.
	Very High	DOT always strictly requires employees to change the username and password of their Mobile Devices.

4. IMPLEMENTATION

As we mentioned in section 3.5 we implement only the Mobile Devices' factors and sub-factors as part of the MFL approach. Figure 6 shows the procedure of Mamdani FIS for Lost Devices Policies. Based on the MF derived from Group 1 and Group 2 we obtain the result of the output variable 'LostDevices' that is a sub-factor of MDM in a Triangular MF format. Figure 7 represents the MF plot for predefined Fuzzy sets in Group 1 that is limited in the range of 0 to 10 and determines each sub-factor's value derived from the security question. The value 10 indicates the highest security for the 'LostDevices' subfactor in Mobile Devices and 0 indicates the lowest one in DOT's network system.

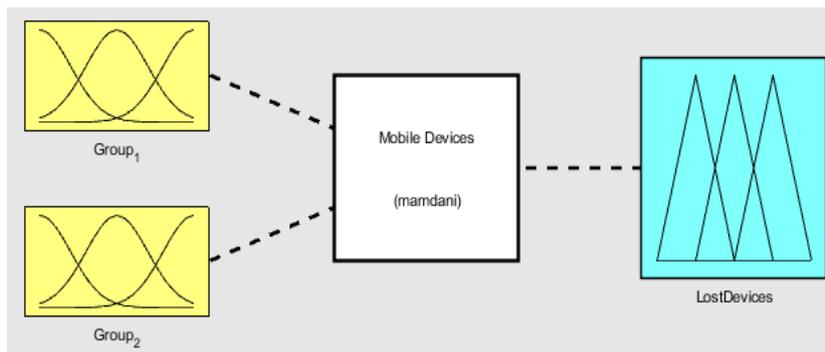


Figure 6. FIS for LostDevices.

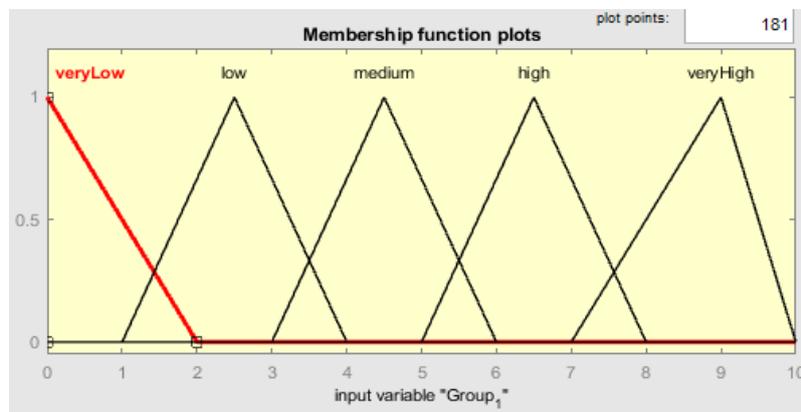


Figure 7. Fuzzy Subset and MF for Group 1.

In the FIS system, before the inference operations, the rules should be defined for the Fuzzy output variable. Thus, we apply *if-then* rules to map the input to the output. The main advantage of *if-then* rules is that they are evaluated in parallel, therefore, the order of rules does not matter. As we mentioned before, these rules are defined based on the expert's answers in the questionnaire that was derived from the DOT's network system. Thus, we define all terms we plan to use in the rules that interpret the values in the input vector and support to assign the appropriate values to the output vector. It is used for a combination of attributes based on the linguistic declaration.

Fuzzy rules are assigned to each group of subsets to provide a functional relationship between Fuzzy attributes. As shown in Figure 8 we defined 25 rules for each group in the Centroid (COA) model. The total number of rules depends on the number of subsets. In this case, we have two

groups and each group has 5 subsets that result in 25 rules. All rules are assigned to the same weight 1, presented at the end of the line for each rule, however, they can be varied in the interval of [0,1].

11. If (Group_1 is medium) and (Group_2 is veryLow) then (LostDevices is low) (1)
12. If (Group_1 is medium) and (Group_2 is low) then (LostDevices is low) (1)
13. If (Group_1 is medium) and (Group_2 is medium) then (LostDevices is medium) (1)
14. If (Group_1 is medium) and (Group_2 is high) then (LostDevices is medium) (1)
15. If (Group_1 is medium) and (Group_2 is veryHigh) then (LostDevices is high) (1)
16. If (Group_1 is high) and (Group_2 is veryLow) then (LostDevices is low) (1)
17. If (Group_1 is high) and (Group_2 is low) then (LostDevices is medium) (1)
18. If (Group_1 is high) and (Group_2 is medium) then (LostDevices is medium) (1)
19. If (Group_1 is high) and (Group_2 is high) then (LostDevices is high) (1)
20. If (Group_1 is high) and (Group_2 is veryHigh) then (LostDevices is high) (1)
21. If (Group_1 is veryHigh) and (Group_2 is veryLow) then (LostDevices is medium) (1)
22. If (Group_1 is veryHigh) and (Group_2 is low) then (LostDevices is medium) (1)
23. If (Group_1 is veryHigh) and (Group_2 is medium) then (LostDevices is medium) (1)
24. If (Group_1 is veryHigh) and (Group_2 is high) then (LostDevices is high) (1)
25. If (Group_1 is veryHigh) and (Group_2 is veryHigh) then (LostDevices is veryHigh) (1)

Figure 8. Fuzzy Rules for LostDevices.

Each of the above if-then rules generates an output in the form of a Fuzzy set. To make a decision based on a single Fuzzy set, we need to apply the aggregation method (Figure 9) to combine all Fuzzy sets from if-then rules to a single Fuzzy set. The last plot in Figure 9 represents the aggregation of all combined Fuzzy sets. In this paper, we use the max-min technique for aggregation and the final value is achieved by the following equation:

$$\text{Final Value} = \max(\text{Group}_1, \text{Group}_2, \dots, \text{Group}_N)$$

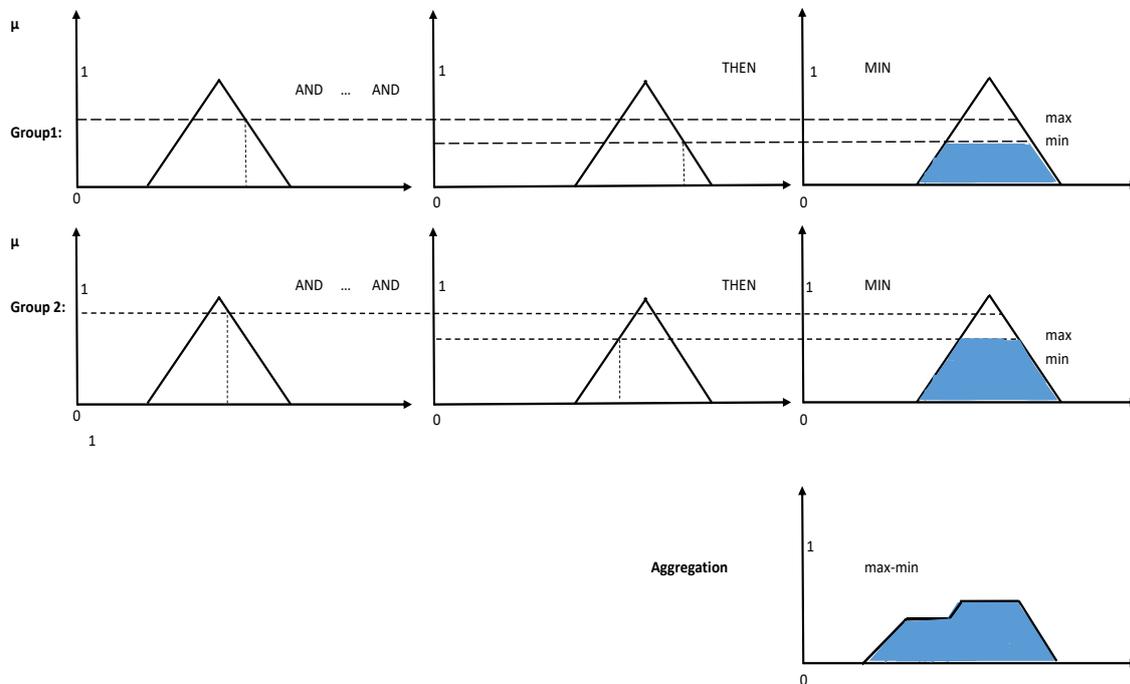


Figure 9. Aggregation in Fuzzy Sets.

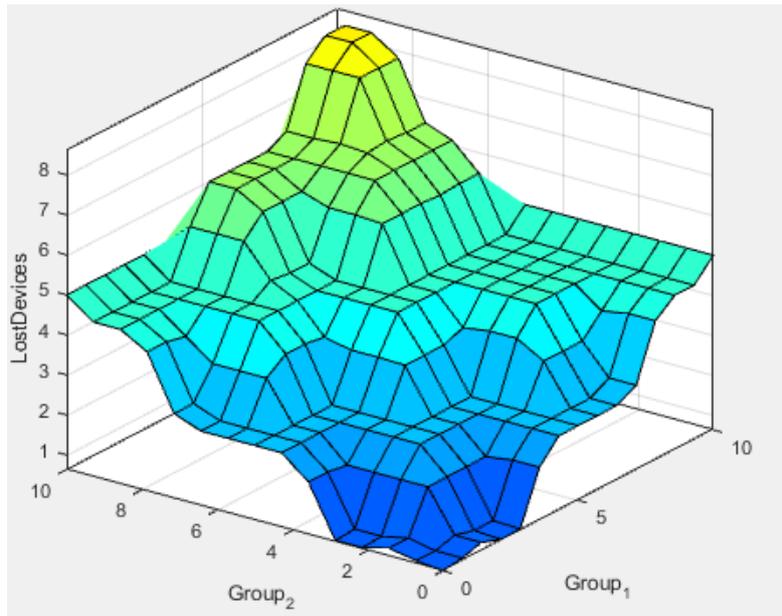


Figure 10. Output Curve for LostDevices.

The three-dimensional curve in Figure 10 depicts the mapping from Group1 and Group 2 to LostDevices. The vertical axis LostDevices represents the range of 0 to 9 that 0 indicates the least security arrangement (maximum vulnerability) for Mobile Devices and 9 indicates the maximum security plan (minimum vulnerability).

Fuzzy Inference processes are presented in Figure 11. It helps us to adjust the input values and obtain the corresponding aggregated output value for each Fuzzy rule. The first two columns of the plot depict the MF referenced by if-part of each rule. This part is called the antecedent and the last column indicates the MF referenced by then-part of each rule which is called the consequent. The average value for inputs and output is displayed on top of each column. Group 1 has an average value of 6.2 and Group 2 has an average value of 7.97 out of 10. The last column displays for LostDevice the result value 6.5 out of 10 (characterized by the last plot of the third column at the very right bottom) that indicates the weighted decision of the inference system based on the aggregation that depends on the input values. The value 6.5 indicates the security of that parameter in Mobile Devices. Therefore, the vulnerability of that is achieved by $10 - 6.5 = 3.5$ up to this point.

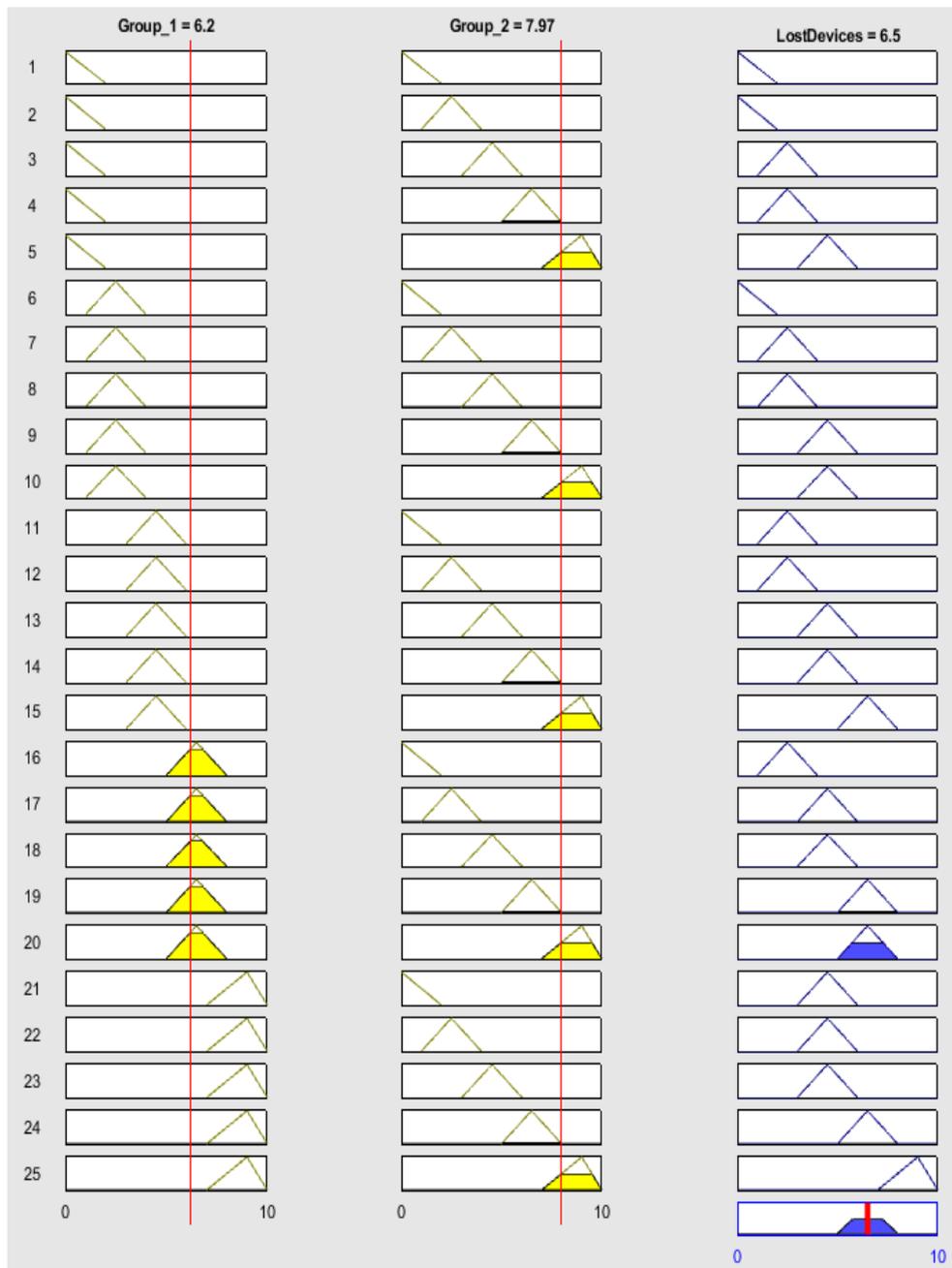


Figure 11. Fuzzy Inference Processes for LostDevices.

In order to achieve the vulnerability of the sub-factor Mobile Device Management System (MDM) as part of the factor Mobile Devices, we need to aggregate the output of the previous step's Fuzzy layer with the output of FIS for Group 3 and 4 in the next layer (Figure 5). As it displays in Figure 12 the output value for this Fuzzy layer is 4.58. Therefore, the output of these two layers is $(6.5+4.58)/2=5.54$ which represents the security level of Mobile Devices at this point. As a result, the vulnerability value for MDM is 4.46. This process will be continued until we obtain the final value for Mobile Devices as part of IoT in DOT.

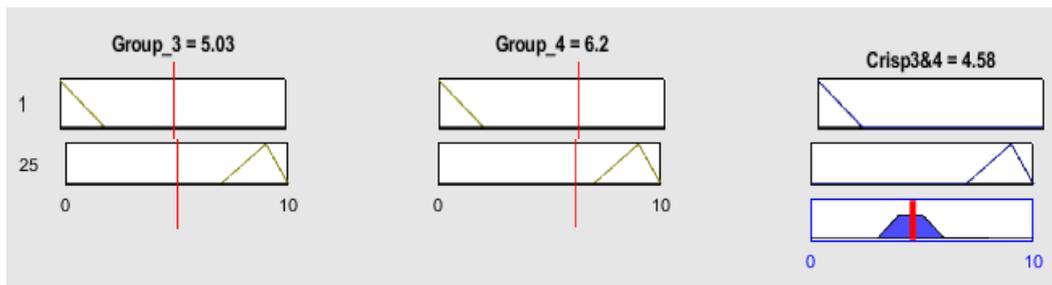


Figure 12. Fuzzy Inference Processes for Group 3 and 4.

5. CONCLUSIONS AND FUTURE WORK

As always network security experts describe the quality of security of a network in a human linguistic manner such as ‘good’ or ‘relatively good’, there would be a variety of interpretations for such description. Thus, our MFL methodology can evaluate quantitative vulnerability values using the Fuzzy Inference System. The findings of this study can be understood as a precise methodology of vulnerability analysis of network security, IoT, etc. in organizations. To our knowledge, this the first report of quantifying vulnerability based on security standards, GQM and without relying on other vulnerability measurement software such as CVSS and Nessus. Broadly translated our findings indicate that this approach can be applied not only in DOT but any agency that wants to measure the vulnerability of their network system quantitatively.

The future work of this research will be sending the questionnaire to the network security experts of DOT and measure the vulnerability of all security aspects of DOT such as physical security, Web Applications, Audit, etc. Moreover, future investigations are necessary to validate the kinds of conclusions that can be drawn from this study.

REFERENCES

- [1] B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan, and K. Trivedi, “Modeling and quantification of security attributes of software systems,” *Proceedings International Conference on Dependable Systems and Networks*, 2002.
- [2] M. U. A. Khan and M. Zulkernine, “Quantifying Security in Secure Software Development Phases,” *2008 32nd Annual IEEE International Computer Software and Applications Conference*, 2008.
- [3] P. Ralston, J. Graham, and J. Hieb, “Cyber security risk assessment for SCADA and DCS networks,” *ISA Transactions*, vol. 46, no. 4, pp. 583–594, 2007.
- [4] C. J. Alberts and A. J. Dorofee, *Managing Information Security Risks: The OCTAVESM Approach*. Addison-Wesley Professional, 2002.
- [5] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, “Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control,” *2007 IEEE Symposium on Security and Privacy (SP 07)*, 2007.
- [6] V. R. Basili and S. Green, “Software Process Evolution at the SEL,” *Foundations of Empirical Software Engineering*, pp. 142–154.
- [7] L. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [8] J. Zhao and B. Bose, “Evaluation of membership functions for fuzzy logic controlled induction motor drive,” *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*.
- [9] H.-J. Zimmermann, “Fuzzy Sets, Decision Making, and Expert Systems,” 1987.
- [10] W. Pedrycz, “Why triangular membership functions?,” *Fuzzy Sets and Systems*, vol. 64, no. 1, pp. 21–30, 1994.
- [11] N. Karnik, J. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.

- [12] W. V. Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy Sets and Systems*, vol. 108, no. 2, pp. 159–178, 1999.
- [13] E. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *International Journal of Man-Machine Studies*, vol. 8, no. 6, pp. 669–678, 1976.
- [14] E. Erturk and E. A. Sezer, "Software fault prediction using Mamdani type fuzzy inference system," *International Journal of Data Analysis Techniques and Strategies*, vol. 8, no. 1, p. 14, 2016.
- [15] A. S. Sodiya, S. A. Onashoga, and B. Oladunjoye, "Threat Modeling Using Fuzzy Logic Paradigm," *Proceedings of the 2007 InSITE Conference*, 2007.
- [16] P. Kamongi, S. Kotikela, M. Gomathisankaran, and K. Kavi, "A methodology for ranking cloud system vulnerabilities," *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013.
- [17] I. V. Anikin, "Information security risk assessment and management method in computer networks," *2015 International Siberian Conference on Control and Communications (SIBCON)*, 2015.
- [18] "Common Vulnerability Scoring System SIG," *FIRST*. [Online]. Available: <https://www.first.org/cvss/>. [Accessed: 03-Feb-2020].
- [19] G. Kuang, X. Wang, and L. Yin, "A fuzzy forecast method for network security situation based on Markov," *2012 International Conference on Computer Science and Information Processing (CSIP)*, 2012.
- [20] KDD Cup 1999 Data. [Online]. Available: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
- [21] "2000 DARPA Intrusion Detection Scenario Specific Datasets," MIT Lincoln Laboratory. [Online]. Available: <https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets>.
- [22] A. Broring, S. Schmid, C.-K. Schindhelm, A. Khelil, S. Kabisch, D. Kramer, D. L. Phuoc, J. Mitic, D. Anicic, and E. Teniente, "Enabling IoT Ecosystems through Platform Interoperability," *IEEE Software*, vol. 34, no. 1, pp. 54–61, 2017.
- [23] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen, "Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach," *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2017.
- [24] "Introducing Nessus," *Nessus Network Auditing*, pp. 27–43, 2004.
- [25] M. Patton, E. Gross, R. Chinn, S. Forbis, L. Walker, and H. Chen, "Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT)," *2014 IEEE Joint Intelligence and Security Informatics Conference*, 2014.
- [26] "Leading the IoT - gartner.com." https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf.
- [27] Boehm, B. W. & Papaccio, P. N. "Understanding and Controlling Software Costs." *IEEE Transactions on Software Engineering SE-4*, 10 (October 1988): 1462-77.
- [28] Ahl V (2005) An experimental comparison of five prioritization methods. Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden.
- [29] V. R. Basili and S. Green, "Software Process Evolution at the SEL," *Foundations of Empirical Software Engineering*, pp. 142–154.
- [30] M. Shepperd, "Practical software metrics for project management and process improvement," *Information and Software Technology*, vol. 35, no. 11-12, p. 701, 1993.
- [31] M. Shepperd, "Practical software metrics for project management and process improvement," *Information and Software Technology*, vol. 35, no. 11-12, p. 701, 1993.
- [32] M. Shojaeshafiei, L. Etzkorn, and M. Anderson, "Cybersecurity Framework Requirements to Quantify Vulnerabilities Based on GQM," SpringerLink, 04-Jun-2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-31239-8_20.
- [33] "The Iso 27001 Risk Assessment," *Information Security Risk Management for ISO 27001/ISO 27002*, third edition, pp. 87–93, 2019.
- [34] H. Susanto and M. N. Almunawar, "Information Security Management Systems," 2018.
- [35] J. T. Force, "Security and Privacy Controls for Information Systems and Organizations," CSRC, 15-Aug-2017. [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/draft>.
- [36] L. Pokoradi and B. Szamosi, "Fuzzy failure modes and effects analysis with summarized center of gravity defuzzification," *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, 2015.
- [37] E. Ngai and F. Wat, "Fuzzy decision support system for risk analysis in e-commerce development," *Decision Support Systems*, vol. 40, no. 2, pp. 235–255, 2005.

- [38] J. Kacprzyk, "Group decision making with a fuzzy linguistic majority," *Fuzzy Sets and Systems*, vol. 18, no. 2, pp. 105–118, 1986.
- [39] A. Lotfi and A. Tsoi, "Learning fuzzy inference systems using an adaptive membership function scheme," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 2, pp. 326–331, 1996.

© 2020 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

A NEW INTELLIGENT POWER FACTOR CORRECTOR FOR CONVERTER APPLICATIONS

Hussain Attia

Electrical, Electronics and Communications Engineering Department,
School of Engineering, American University of Ras Al Khaimah,
Ras Al Khaimah, UAE

ABSTRACT

This paper presents a new design of a unity power factor corrector for DC-DC converter applications based on an Artificial Neural Network algorithm. The controller firstly calculates the system power factor by measuring the phase shift between the grid voltage and the grid current. Secondly, the controller receives the absolute value of the grid voltage and the measured phase shift through the designed ANN, which predicts the duty cycle of the pulse width modulation (PWM) drive pulses, these PWM pulses are used to drive the Boost DC-DC converter to enforce the drawn current to be fully in phase with the grid voltage as well as to improve the level of Total Harmonics Distortion (THD) of the grid current. MATLAB/Simulink software is adopted to simulate the presented design. The analysis of the simulation results indicates the high performance of the proposed controller in terms of correcting the power factor, and improving the grid current THD.

KEYWORDS

Power factor corrector, artificial neural network, Boost DC-DC converter, Total Harmonic Distortion, MATLAB/Simulink.

1. INTRODUCTION

The usage of DC-DC converter in homes and industrial appliances leads to negative effects on the connected grid due to the non-linear and the reactive components of the electrical system. The negative effects are represented by increasing the phase shift of the grid current with respect to the grid voltage and the waveform distortion of the grid current. These effects are increasing the electrical losses due to the low system power factor in addition to increasing the Total Harmonic Distortion of the grid current due to deteriorating current waveform shape. From this fact, enhancing the power factor, which is the rate of the active power to the apparent power, becomes necessary to reduce energy loss and to enhance the THD level of the current waveform. In other words, the unity power factor of the connected loads is necessary to effectively reduce energy loss in the electrical system's components through reducing the reactive power absorption. On the other hand, the inductive, the capacitive, and the other non-linear components are negatively affecting the waveform shape of the drawn current from the grid, this distorts the current waveform and increases the Total Harmonics Distortion (THD) level [1]-[5]. Many research studies have been proposed for the objective of enhancing the rate of active power to apparent power reaching to the target of unity power factor to guarantee the removal of energy loss and to improve the level of the current THD [6]-[12].

For a bridgeless Cuk converter, the study [1] proposed a novel power factor corrector (PFC) for better power factor performance and for lower total harmonic distortion of the grid current. The converter supplied power for the LED lamp applications, reduced the losses and improved the converter efficiency. The work in [2] presented a design, operation's principle, and simulation results analysis of a PFC through a variable switching frequency pulse width modulation controlling technique. The proposed PFC controlling system enhanced the THD, and power system efficiency through a new PFC controller. The system worked on absorbing a sinusoidal current waveform that reduced the current THD, and also regulated the load voltage. The effect of the total harmonic distortion during the power factor correcting has been shown and analysed in [3] through a non-linear load. The sinusoidal waveform of the grid current is affected by the type of the connected non-linear loads that increase the harmonic distortion. The system performance has been analysed in the study in terms of the THD level.

The work in the studies [4]-[5] presented a simple power factor controller based on micro controllers; an accurate automatic single phase power factor controller has been implemented using the micro controller type ATmega328 in [4]. Whereas, the same micro controller has been adopted in [5] to produce the PWM drive pulses to the boost converter for the power factor correcting function.

The Fuzzy logic controller (FLC) had been proposed in different researches for the objective of power factor correction [6]-[8]. The study [6] has proposed a simulation study with results analysis of a Fuzzy Logic Controller FLC based electrical power system for power factor controlling. By controlling the excitation current of the inserted synchronous motor, the proposed FLC controller compensated the requirement of the mathematical models which is needed for parameter determining of a conventional power factor control. An analytical study for correcting and enhancing the power factor through a boost converter has been presented in [7] based fuzzy logic controller, proportional-integral controller, as well as a hysteresis current controller, in study [7], the voltage loop controller has been represented by the fuzzy controller and the PI controller. Whereas, two controllers have been designed and simulated in [8] for a single phase power factor correcting function. Fuzzy logic controller and predictive controller have been presented in the study for the power factor correction aim.

2. POWER FACTOR CORRECTION BASED ON ARTIFICIAL NEURAL NETWORKS IN LITERATURE

Artificial Neural Networks (ANN) have been used for different problems solving, and currently ANN have been proposed to improve different systems' performance. ANN has been adopted for the power factor improvement as an effective power factor corrector [9]-[12]. An ANN has been presented in [9] through a novel technique for a line power factor correcting via a synchronous motor. The motor is directed by the ANN controller to have robust system of improved PF (0.984) through compensating the reactive power of the system under the test.

The study of [10] offered a developed power factor corrector for a certain power system. A clustering neural network has been adopted in the study to achieve effective performance in terms of load varying. The presented PFC of [10] optimized the power factor automatically ranging between 0.9 and 0.93. In [11], a real-time reactive power corrector has been presented based on ANN. The ANN received the active and reactive power of each load and based on the loads' values and types, the output of the ANN controlled the switching ON/OFF of the system's capacitors. The testing step of the study reflected the satisfied compensating of the reactive power with power factor records in a range of 0.94 to 1. A comparative study has been presented in the work of [12] with respect to the power factor correcting performance, the effectiveness of

an ANN-based controller has compared with a conventional PI controller in terms of power quality.

Based on the above studies' findings, which show the effectiveness of the PFC controllers based on artificial neural network, this paper proposes an effective new power factor correction controller for a DC-DC converter applications based on a designed algorithm of an artificial neural network. The remaining of this paper is arranged as follows; Section 3 explains the methodology of the proposed power factor corrector based on neural network, Section 4 explains the proposed system, Section 5 shows the simulation results with results analysis, whereas Section 6 concludes the study findings and focuses on aspects of future work.

3. METHODOLOGY OF THE PROPOSED POWER FACTOR CORRECTOR BASED ON ARTIFICIAL NEURAL NETWORK

The types of the connected electrical loads to the grid are divided into active loads, reactive loads, which are either inductive or capacitive loads, and non-linear loads, which are represented by the semiconductor devices [6]. The active loads transform the grid's energy to useful energy. Whereas, the reactive loads (inductive or capacitive) are absorbing the grid energy without transforming it to useful energy, and also shifting a certain lag or lead phase of grid current with respect to grid voltage. Non-linear devices are distorting the waveform of the grid current which leads to increase the THD level of the load current [13]-[14]. Normally, the value of the power factor (PF), which is shown in (1), is less than 0.85 with respect to the inductive or capacitive loads.

$$PF = \frac{\text{Active Power}}{\text{Apparent Power}} = \frac{P}{S} \quad (1)$$

Where P is the active power in Watt (W), S is the apparent power in Volt·Amp (V·A), which is a vector value, as shown in (2).

$$S = P + jQ \quad (2)$$

Where Q is the reactive power in Volt·Amp (V·A). Figure 1 shows the case of active, reactive, and apparent power for the case of inductive load. From Figure 1, the power factor PF relation can be represented by (3).

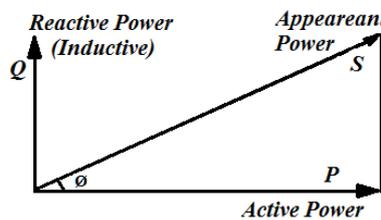


Figure 1. Vectors of active, reactive, and apparent power for inductive load condition

$$PF = \frac{P}{S} = \cos(\phi) \quad (3)$$

From (3), to have a better PF , the phase shift ϕ should be in lower value toward zero for the case of unity power factor. In other words, to have a high or a unity power factor, the phase shift between grid voltage and drawn current should be near zero for high PF , or exactly zero to

obtain unity *PF*. So, for the case of low *PF* inductive load, certain capacitors are necessary to improve the value of *PF*.

In the case of DC-DC converter, the instantaneous values of the PWM pulses are playing the major role to compensate the undesirable phase shift. From this point of problem solving, the neural network is adopted to predict the required compensation of the instantaneous phase shift.

Through a machine learning algorithm of neural network, a fast and accurate response can be guaranteed [15]-[17]. Big size of numerical data of input and output variables is necessary for the learning or training process. The structure of any ANN algorithm includes one input layer, one output layer, and one or more hidden layers. Figure 2 shows the basic unit of this algorithm represented by the neuron. The functions of the neuron are, firstly, adding the input variables (X_1 to X_n) after the weighting step of each input variable (W_1 to W_n) as shown in (4), secondly, passing the addition result to an activation function. This function can be one of the three activation functions, namely, the linear (5), the sigmoidal (6), and the hyperbolic (7) transfer functions respectively.

$$Z = \sum_{i=1}^N W_i X_i + B \quad (4)$$

$$y_n = f(z) \quad (5)$$

$$f(z) = \frac{1}{1 + \exp^{-z}} \quad (6)$$

$$f(z) = \frac{1 + \exp^{-2z}}{1 - \exp^{-2z}} \quad (7)$$

The learning process and weight updating are done based on the back propagation process, and the accuracy of the designed ANN is evaluated by the value of the mean square error (*MSE*). This evaluating parameter represents the difference between the target value and the estimated value by the designed ANN as shown in (8).

$$MSE = \frac{1}{q} \sum_{k=1}^q [t(k) - a(k)]^2 \quad (8)$$

Where $t(k)$ is the target, $a(k)$ is the ANN estimated value, and q is the element number of the input variables.

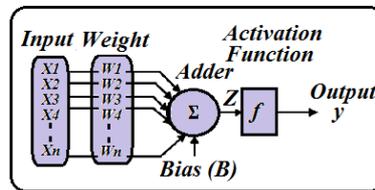


Figure 2. The structure of the neuron

4. PROPOSED POWER FACTOR CORRECTION CONTROLLER FOR BOOST DC-DC CONVERTER

The proposed controller works on effectively increasing the power factor of the DC-DC converter with its load to unity *PF* and improving the Total Harmonic Distortion (THD) of the

grid drawn current through the ANN predicting function for the required opposite phase shifting based on the actual phase shift of the grid current. The presented system block diagram is shown in Figure 3. Two parts are involving in the system; the power part and the controlling part. The power part includes a rectifier, a DC-DC converter, and a resistive load. The second one is a controlling part which includes a phase shift and power factor calculator, a neural network algorithm, and a pulse width modulation drive pulses generator. The proposed controller receives the instantaneous values of the grid voltage waveform, and grid drawn current, these two sensed values are used to calculate the phase shift and power factor. The other received values by the controller are the instantaneous values of load voltage and load current, these values are needed for the controlling function.

Figure 4 shows the designed ANN of one input layer, one hidden layer, and one output layer. The algorithm includes 2 neurons in the input layer, 25 neurons in the hidden layer, and 1 neuron in the output layer. Absolute value of the grid voltage, and phase shift between grid voltage and grid current are received by the input layer of the algorithm. After the processing of the neural network, the predicted duty cycle (DI) is estimated and produced at the terminal of the output layer.

The drive pulses generator adopts the process of the pulse width modulation technique [18]-[23], this generator produces PWM drive pulses of switching frequency 10 kHz, the duty cycle D of these pulses is controlled based on the sensed load voltage, load current, and the predicted value on DI , which is produced by the designed ANN algorithm. The parameters of the boost DC-DC converter are designed based on the studies of [24], and Table 1 shows the setting of these parameters.

5. SIMULATION RESULT ANALYSIS

MATLAB Simulink software is applied to simulate the proposed power factor corrector for DC-DC converter applications. The records are collected firstly with fixed duty cycle ($D = 0.6$) with two values of load resistors as shown in Table 1. Secondly, the system performance is evaluated through the proposed power factor controller. The first step in simulation after designing the ANN algorithm with one input layer of 2 neurons, one hidden layer of 25 neurons, and one output layer of 1 neuron simulates the algorithm as shown in Figure 5 (a). The performance of the designed and simulated ANN algorithm reflects the best validation of 0.00096955 at the epoch 4 as shown in Figure 5 (b) in terms of estimating the instantaneous value of duty cycle.

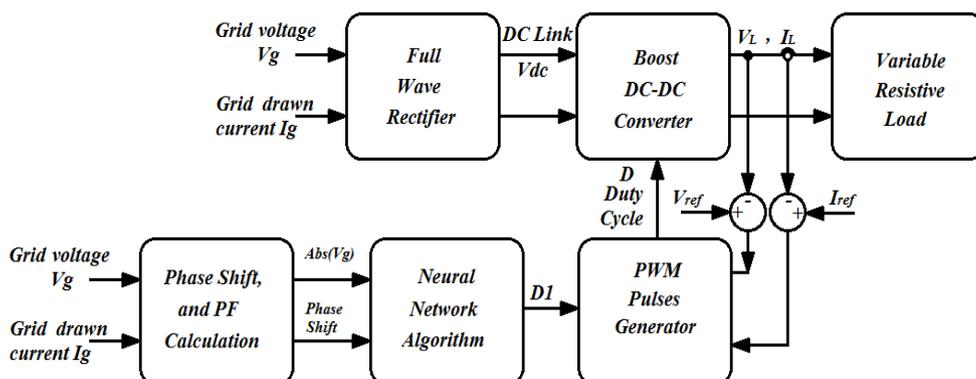


Figure 3. The proposed system block diagram

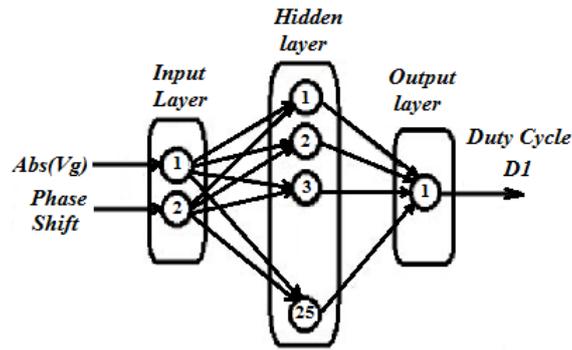
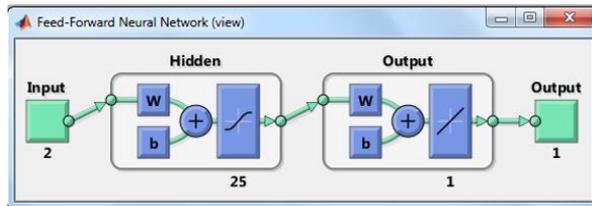


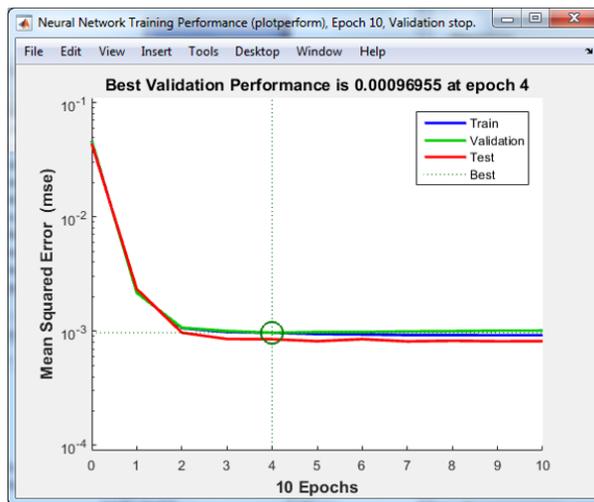
Figure 4. The designed ANN algorithm

Table 1. Converter parameters

Parameter	Value
Inductor	10 mH
Capacitor	1000 μ F
Load Resistor	75 Ω , 150 Ω
Switching Frequency	10 kHz
Duty Cycle(without control)	0.6



(a)



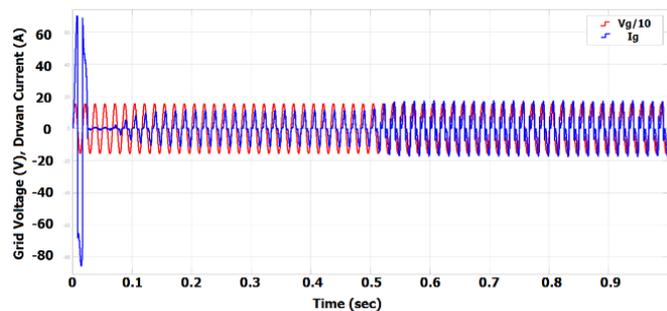
(b)

Figure 5. The designed and simulated ANN algorithm, (a) Feed-forward ANN structure, (b) ANN performance

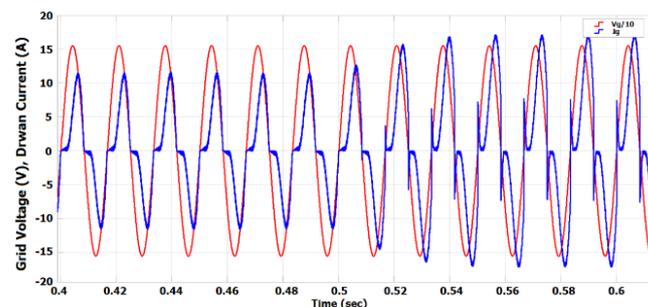
Figure 6 shows the system performance with fixed duty cycle ($D = 0.6$) in terms of the grid drawn current with respect to the grid voltage at two different loads, the distortion of grid current is clearly noticeable in Figure 6 (b). Figure 7 shows the harmonics spectrum of the grid current in which high levels of low order harmonics and THD of 44.76% are monitored. Figure 8 shows the uncontrolled load voltage during 1 sec simulation period which is divided into two equal periods. In the first period of 0.5 sec, the connected load is 150Ω , whereas the second period is with load of 75Ω . From Figure 6 to Figure 8, the distortion in the shape of grid current waveform as well as the marked phase shift between the grid voltage and the current are clearly noted, the phase shift between the current and the voltage, and distorted current all leading to having low power factor and high Total Harmonics Distortion THD in current waveform respectively.

The system performance through the proposed controller is monitored through Figure 9 to Figure 11. Figure 9 shows the grid current at the same considered loads, the lower level of grid current distortion is shown in Figure 9 (b). Figure 10 shows the harmonics spectrum of the grid current which involves the lower levels of low order harmonics, in addition to lower THD level. Figure 11 shows the load voltage during the same 1 sec simulation period with the same resistive loads consideration. Figure 11 proves the high priority of the proposed controller in terms of controlling the load voltage, softening the starting and transient response compared to the response of the fixed duty cycle of Figure 8.

Table 2 shows load voltage, power factor, and THD at the two resistive loads. The effectiveness of the presented controller is clear in the terms of the mentioned parameters improvement. Table 3 compares the performance of the proposed controller with respect to previous controllers in literature in terms of the power factor. The contents of Table 3 reflect the high priority of the presented controller in terms of the level and the stability of the recorded power factor.



(a)



(b)

Figure 6. Grid drawn current with grid voltage, 0 to 0.5 sec (load= 150Ω), 0.5 to 1 sec (load= 75Ω); (a) I_g and V_g for full 1 sec simulation time, (b) Zoom in for the steady state waveforms.

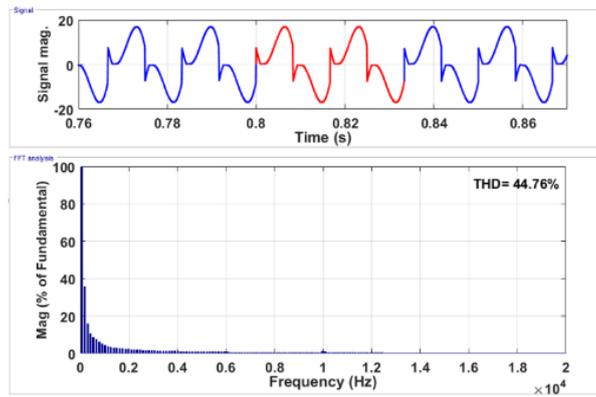


Figure 7. Harmonic spectrum of the dawn current at load $R_L=75 \Omega$

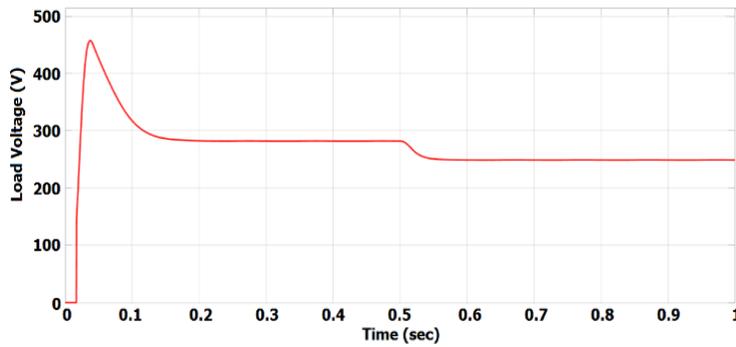
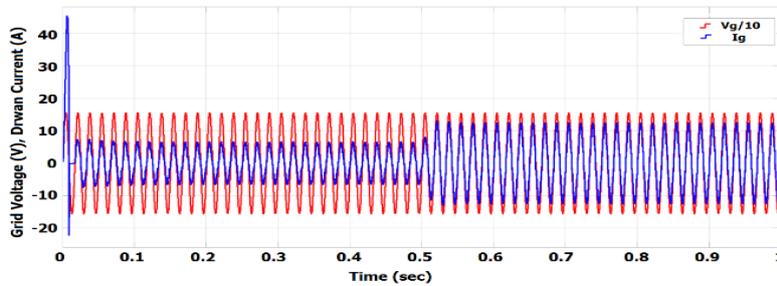
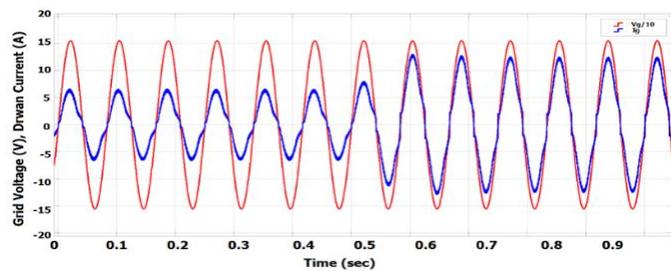


Figure 8. Load voltage during 0 to 0.5 sec at load 150Ω , and during 0.5 to 1 sec at load 75Ω .



(a)



(b)

Figure 9. Grid current and the grid voltage using the proposed PFC controller, 0 to 0.5 sec (load= 150Ω), 0.5 to 1 sec (load= 75Ω); (a) I_g , and V_g , (b) Zoom in of I_g , and V_g

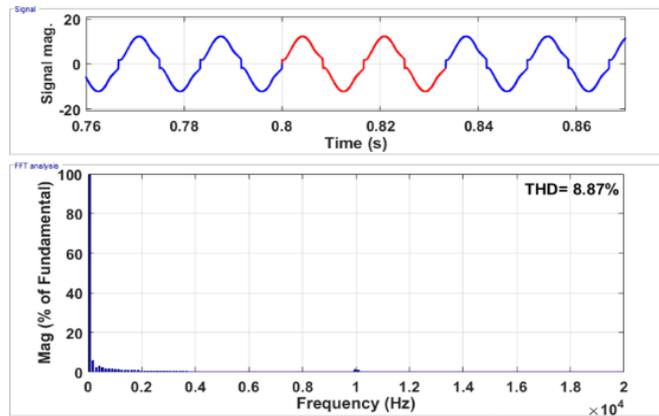


Figure 10. Harmonic spectrum of the dawn current using the proposed PFC controller at load $R_L=75 \Omega$

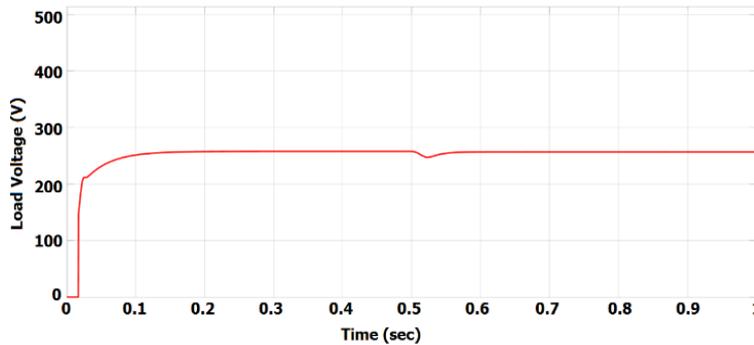


Figure 11. Load voltage using the proposed PFC controller; (0 to 0.5 sec) at load 150Ω , and (0.5 to 1 sec) at load 75Ω .

Table 2. Load voltage, power factor, and THD records with/without control at 2 different loads

Load	Load Voltage (V)		Power Factor		THD	
	$D = 0.6$	Controlled D	$D = 0.6$	Controlled D	$D = 0.6$	Controlled D
$R_L = 75 \Omega$	249 V	257 V	0.8314	0.9997	44.76%	8.87%
$R_L = 150 \Omega$	282 V	257 V	0.8124	0.9998	38.91%	12.88%

Table 3. Power factor comparison

	Presented PF controller	PF controller of [9]	PF controller of [10]	PF controller of [11]
Power Factor	0.9997 0.9998	0.984	0.9 \rightarrow 0.93	0.94 \rightarrow 1

6. CONCLUSION AND FUTURE WORK

The study presented an effective power factor correction controller, the proposed controller is designed based on an algorithm of neural network for converter applications. The controller guarantees three aims. The first one is highly improving the power factor of the system. The second aim is enhancing the low order harmonics and level of the grid current THD, whereas the third aim is controlling the load voltage and removing the overshoot of load voltage at starting

instants. The collected records showed the increasing PF from less than 0.85 to approximately unity PF at different loads. The enhancing of the THD and low order harmonics of grid current are also gained, as well as stabilizing the load voltage at 257 V compared to the unstable load voltages of 249 V, and 282 V at the load's resistance of 75 Ω , and 150 Ω respectively.

In terms of future work, the function of the neural network algorithm could be extended to include the total controlling objectives involving the voltage controller, in addition to the proposed function of ANN in this study.

ACKNOWLEDGEMENTS

The author appreciates the school of engineering in American University of Ras Al Khaimah for the financial support.

REFERENCES

- [1] Saravanan D, Gopinath M, (2016), "A Novel Power Factor Correction Modified Bridge Less-CUK Converter for LED Lamp Applications" *IJPEDS*, Vol. 7, No. 3, pp. 880-891.
- [2] Malliseti R. K., Duraisamy L., Ch Sai B., (2011), "A Variable Switching Frequency with Boost Power Factor Correction Converter", *TELKOMNIKA*, Vol. 9, No. 1, pp. 47-54.
- [3] R. A. Rani, S. Saat, Y. Yusop, H. Husin, F. K. Abdul Rahman, A. A. Isa, (2016), "The Effects of Total Harmonics Distortion for Power Factor Correction at Non-Linear Load", *IJPEDS*, Vol. 7, No. 2, pp. 543-550.
- [4] Nurul H. I., Muhammad N. Z., Nur A. S., Faizal M. T. T., Aini H. M. S., (2015), "A design of an Automatic Single Phase Power Factor Controller by using Arduino Uno Rev-3", *Applied Mechanics and Materials*, Vol. 785, pp 419-423.
- [5] Kartikesh K. J., Bidyut M., Prem P., Kartick C. J., (2016), "Hardware Implementation of Single Phase Power Factor Correction System using Micro-Controller", *IJPEDS*, Vol. 7, No. 3, pp. 790-799.
- [6] Ahmet G., Ö. Fatih K., Hakan A., Ceyhun Y., Mustafa Ş.i, (2016), "Simulation Study on Power Factor Correction Controlling Excitation Current of Synchronous Motor with Fuzzy Logic Controller", *IJISAE*, 4(Special Issue), pp. 229–233.
- [7] A. Bhakthavachala, S. Tara kalyani, K. Anuradha, Nukala Vengaiyah, (2017) "Fuzzy Logic Controller based Unity Power Factor Correction of Boost Converter", *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, Volume 12, Issue 6 Ver. II, PP 52-58.
- [8] Abdelhalim Kessal, (2014), "Combined Fuzzy and Predictive Controller for Unity Power Factor Converter", *International Journal of Electrical and Computer Engineering*, Vol:8, No:9, 2014, pp. 1541-1546.
- [9] S. Sagioglu, I. Colak B, R. Bayindir, (2006) "Power factor correction technique based on artificial neural networks", *Energy Conversion and Management*, Vol. 47, pp. 3204–3215.
- [10] Afaf A. A., (2018), "Improved Power Factor of Electrical Generation by using Clustering Neural Network", *IJAER*, Vol. 13, No. 7, pp. 4633-4636.
- [11] Carl John O. Salaan and Noel R. Estoperez, (2011), "An Artificial Neural Network Based Real-time Reactive Power Controller", *Proceedings of the World Congress on Engineering and Computer Science*, Vol. I, WCECS 2011, October 19-21, 2011, San Francisco, USA.
- [12] N. Ramchandra, M. Kalyanchakravarthi, (2012), "Neural Network Based Unified Power Quality Conditioner", *International Journal of Modern Engineering Research (IJMER)*, Vol.2, Issue.1, pp-359-365.
- [13] Masashi O., Hirofumi M. (2003), "An AC/DC Converter with High Power Factor", *IEEE Trans. on Industrial Electronics*, Vol. 50, No. 2, pp. 356–361.
- [14] M. Malinowski, M. Jasinski, M.P. Kazmierkowski, (2004) "Simple direct power control of three-phase PWM rectifier using space-vector modulation (DPC SVM)", *IEEE Trans. on Industrial Electronics*, pp. 447–454.
- [15] Hussain A. Attia, (2019), "High performance PV system based on artificial neural network MPPT with PI controller for direct current water pump applications, *IJPEDS*, Vol. 10, No. 3, pp. 1329-1338.

- [16] Hussain Attia, (2019), "Supplying DC Electricity to the Isolated Dwellings through MPP Tracked PV System Based on Artificial Neural Network", 6th International Conference of Control, Dynamic Systems, and Robotics (CDSR'19), Ottawa, Canada.
- [17] Hussain Attia, (2018), "Artificial Neural Networks Based Maximum Power Point Tracking Photovoltaic System for Remote Parks LED Lighting Applications", International Review on Modelling and Simulations, Vol. 11, Iss. 6, pp. 396-405.
- [18] H. A. Attia, T. K. S. Freddy, H. S. Che, W. P. Hew, A. El Khateb, (2017), "Confined Band Variable Switching Frequency Pulse Width Modulation (CB-VSF PWM) for Single-Phase Inverter with LCL Filter," IEEE Trans. on Power Electronics, Vol. 32, No. 11, pp. 8593–8605.
- [19] Hussain Attia, (2018), "A Stand-alone Solar PV System with MPPT Based on Fuzzy Logic Control for Direct Current Portable House Applications", IREMOS, Vol. 11, Iss. 6, pp. 377-385.
- [20] Hussain Attia, (2018), "Fuzzy Logic Controller Effectiveness Evaluation through Comparative Memberships for Photovoltaic Maximum Power Point Tracking Function", IJPEDS, Vol. 9, No. 3, pp. 1147-1156.
- [21] Attia Hussain, Del Ama G. F., (2019), "Stand-alone PV System with MPPT Function Based on Fuzzy Logic Control for Remote Building Applications", IJPEDS, Vol. 10, Iss. 2.
- [22] Hussain A Attia, (2017), "Comparative design of fuzzy logic controller for photovoltaic panel maximum power point tracking", ICECTA2017, Ras Al Khaimah-UAE.
- [23] Hussain Attia, Amjad Omar, Maen Takruri, Halah Y. Ali, (2017), "Pulse Width Modulation Based Decentralized Street LED Light," International Journal of Power Electronics and Drive System (IJPEDS), vol. 8(3), pp. 1238-1247.
- [24] Hussain A. Attia, Y. I. Al-Mashhadany, Beza N. G., (2014), "Design and Simulation of a High Performance Standalone Photovoltaic System," ICREGA'14 – Renewable Energy: Generation and Applications, Springer Proceedings in Energy 2014.

AUTHOR

Hussain Attia has Ph.D. degree in Power Electronics. Research interests: Power Electronics System, AC & DC Speed Control Drives, PWM Inverters and harmonics reduction techniques.



A FLOW SIMULATION IN THE FOAMING PROCESS

Karel Fraňa¹, Jörg Stiller² and Iva Nová¹

¹Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic

²Technische Universität Dresden, Institut für Strömungsmechanik
01062, Dresden, Germany

ABSTRACT

This paper deals with unsteady three-dimensional numerical calculations of the two-phase flow problem represents a gas bubble formation in the liquid in the container with the specific size. For calculations, the Volume of Fluids approach is adopted to resolve the shape of bubbles and their dynamics. The liquid phase is a mixture of water-ethanol and the gas phase is considered as air. The problem is treated as isothermal. The study is still limited to the lower flow rates at which bubbles are created and rising separately without any interaction, merging etc. However, this particular problem still required finer mesh especially in the domain in which bubbles are formed. The current results showed that air bubbles have a form of the ellipsoid and after they reach the liquid surface, they are moving towards to the side walls along this liquid level. This fact is interesting from the view of the foaming process and for the other investigation of the bubble behaviour at this phase interface. The flow study is calculated parallel using compressible multi-phase flow solver.

KEYWORDS

Multi-phase flow simulations, Computational mesh, Parallel calculations & Flow visualisation

1. INTRODUCTION

The numerical calculation of the multi-phase flow problem is an attractive approach to predict flow behaviours in technical applications. Generally, there are a lot for examples where it is possible to observe an interaction between liquid and gas phases in various equipment designs. If this flow feature of the multi-phase behaviour is known in advance due to numerical simulations, a design of real equipment can be improved. Consequently, it can save time and finally cost in the production phase of the prototype examination. Another problem, where the theoretical calculation of the multiphase flow problem can be applied, is a forming process used in the production of foam materials. There are several techniques to manufacture foams and the study is focused on the process in which the metal foams are directly fabricated by injecting a gas in molten metal alloys. To control this foaming process, the knowledge about bubble behaviour in the liquid is essential. If the temperature effects and solidification process should be considered as well, the problem becomes to be really complex [6]. Nowadays there are several ways how to calculate liquid-gas interface e.g. Volume of Fluids (VOF) approach, the immersed boundary method (IBM) etc. VOF method uses a function which denotes the fractional amount of particular fluid in at spatial location at time t . The advection term in equations is solved by pseudo-Lagrangian geometrical advection schemes in order to minimize numerical diffusion. This fact makes VOF [3] the most mass conservative among all multi-fluid interface tracking methods. This method was successfully applied for a problem of the bubbles generation and

rising bubbles in [2]. The two-phase flow over two types of step-poolspill was successfully investigated using two-phase schemes VOF and with various turbulence modelling in [5]. In contrast IBM uses non-body conformal structured Cartesian grids for simulating flow through complex geometries. The main advantages of IBM are simplicity in grid generation and discretization of the Navier-Stokes equations. So that it is easier to develop a code, less memory requirement to store the grid information and higher computational efficiency. In [1] an application of the IBM for a single rising bubble was discussed. This paper is focused on the investigation of rising gas bubbles in stagnant liquid. To obtain experimental results such as velocities, pressure, structure shape etc. will be generally difficult in aluminium alloys. The simulation study was carried out in the water-ethanol mixture with gas. Therefore obtained numerical results can be later validated by experiments. The paper is organised as follows: in chapter 2, the mathematical model, computational mesh and experimental background are introduced. Several founded results are presented and discussed in chapter 3. Chapter 4 contains conclusion to results and further scheduled work is briefly illuminated.

2. PROBLEM FORMULATION

2.1. Mathematical Model

The essential point of the foaming process e.g. by the continuous gas injection is a doping of stabilization components into the melted alloys which cause a change of the material properties influencing further the flow features. This fact is a subject of the flow investigation. The gas injection is represented by air. The liquid phase is a mixture of water-ethanol which in fact represents aluminium alloys. The interface between gas and liquid fraction represents typically the two-phase flow problem. The three dimensional numerical simulation in the container is carried out in OpenFoam with the “CompressibleMultiPhaseInterFoam” Solver. The “heRhoThermo” model is used typically for the fixed mixture composition. For the gas fraction, the perfect gas equation is used in a form as follows,

$$\rho = \frac{1}{RT} p, \quad (1)$$

where ρ is air density, p pressure, T temperature and R is gas constant. The compressibility of the gas phase is taken into account. There are several works that supposed the compressibility can be neglected under some specific circumstances e.g. in [12]. The fluid phase is considered to be perfect fluid with equation in (2),

$$\rho = \frac{1}{RT} p + \rho_0, \quad (2)$$

where ρ_0 is the density at the temperature $T=0$. Material mixture properties are prescribed using the volume fraction function e.g. for the phase 1, which takes a form as

$$\alpha_1 = \frac{\Omega_1}{\Omega_1 + \Omega_2 + \Omega_3} \quad (3)$$

where Ω_1, Ω_2 and Ω_3 are volume fraction functions of the phase 1, 2 or 3, respectively. Similarly, the volume fraction function Ω_2 can be formulated as in Eq.(3) or directly calculated using $\alpha_2 = 1 - \alpha_1 - \alpha_3$. A conservation law of mass for each component is defined as well. The equation of the mass conservation can be defined for 3 different phases taking a form as in Eq. (4), (5) and (6).

$$\frac{\partial \alpha_1 \rho_1}{\partial t} + \frac{\partial}{\partial x} (\alpha_1 \rho_1 u) + \frac{\partial}{\partial y} (\alpha_1 \rho_1 v) + \frac{\partial}{\partial z} (\alpha_1 \rho_1 w) = 0 \quad (4)$$

$$\frac{\partial \alpha_2 \rho_2}{\partial t} + \frac{\partial}{\partial x} (\alpha_2 \rho_2 u) + \frac{\partial}{\partial y} (\alpha_2 \rho_2 v) + \frac{\partial}{\partial z} (\alpha_2 \rho_2 w) = 0 \quad (5)$$

$$\frac{\partial \alpha_3 \rho_3}{\partial t} + \frac{\partial}{\partial x} (\alpha_3 \rho_3 u) + \frac{\partial}{\partial y} (\alpha_3 \rho_3 v) + \frac{\partial}{\partial z} (\alpha_3 \rho_3 w) = 0 \quad (6)$$

The velocity vector is given by u, v and w components. Density of each phase is denoted by ρ_1 , ρ_2 or ρ_3 . The equation of momentum takes a form as in Eq. (7), (8) and (9).

$$\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} (\rho u^2 + p) + \frac{\partial}{\partial y} (\rho uv) + \frac{\partial}{\partial z} (\rho uw) = F_x \quad (7)$$

$$\frac{\partial}{\partial t} (\rho v) + \frac{\partial}{\partial x} (\rho uv) + \frac{\partial}{\partial y} (\rho v^2 + p) + \frac{\partial}{\partial z} (\rho vw) = F_y \quad (8)$$

$$\frac{\partial}{\partial t} (\rho w) + \frac{\partial}{\partial x} (\rho wu) + \frac{\partial}{\partial y} (\rho vw) + \frac{\partial}{\partial z} (\rho w^2 + p) = F_z \quad (9)$$

F_x , F_y and F_z are components of the external forces e.g. surface tension forces or gravity force. The mixture density ρ is used for all three equations (7), (8) and (9). The energy equation for temperature of the mixture is given in Eq. (10).

$$\frac{\partial \rho T}{\partial t} + \frac{\partial}{\partial x} (\rho u T) + \frac{\partial}{\partial y} (\rho v T) + \frac{\partial}{\partial z} (\rho w T) = W_p \quad (10)$$

W_p represents a pressure work which must be included if a compressible flow is expected. For definition of different flow state in regards to the bubble shape, several non-dimensional parameters are commonly used. The Eötvös number in Eq. (11) expresses the ration of the body force to the surface tension force,

$$Eo = \frac{\rho_0 g D^2}{\sigma} \quad (11)$$

Where ρ_0 is density of the continuous phase (in our case liquid mixture) and σ is interfacial surface tension between gas and liquid phase. Parameter D is a characteristic length relating to the characteristic diameter of the formed bubble. The dynamics of bubble is influenced by the buoyancy force to the viscous force given by the Galilei number.

$$Ga = \frac{\rho_0 g^{1/2} D^{3/2}}{\mu_0} \quad (12)$$

In Eq. (12), the dynamics viscosity μ_0 is related to continuous liquid phase. The Morton number can be calculated based on the Eo and Ga numbers as is denoted in Eq. (13).

$$Mo = \frac{Eo^3}{Ga^4} \quad (13)$$

Relevant parameters Eo or Ga contains an information about bubble diameters, however, bubble dynamics is missing. In order to involve the bubble dynamics expresses by bubble velocities, bubble Reynolds number (15) and Weber number (14) can be used.

$$We = \frac{\rho u^2 D}{\sigma} \quad (14)$$

$$Re = \frac{\rho u D}{\mu} \quad (15)$$

Reynolds and Weber parameters can take into account the influence of the terminal velocity which can be found easily using numerical simulation. The terminal velocity can describe the unsteady bubble dynamics as well [11].

2.2. Setting of Calculations

For the iterative process of pressure quantity, a preconditioned conjugate gradient method is applied. Other quantities e.g. temperature or velocity etc. are iteratively calculated using Gauss Seidel schemes. Unsteady terms of mathematical models are treated using Euler scheme providing the accuracy of the first order. For the gradient operator, the Gauss linear scheme is applied; Laplacian operator is treated by Gauss linear uncorrected scheme and in the divergence operator of the most quantities Gauss linear scheme is adopted, however, due to specific instabilities in the calculation, for the velocities a Gauss upwind scheme had to be prescribed. Relaxation factors varied between 0.95 and 0.9 were applied for calculation of the velocity and pressure quantities.

2.3. Computational Domain and Mesh

The computational domain is determined by a real container with a size of $0.2\text{m} \times 0.2\text{m} \times 0.2\text{m}$. The liquid phase level (water-ethanol mixture) is in the height of 0.12m . The rest of the computational domain represents the gas phase. The grid is refined at the place of the gas injection, at the container bottom, in the area of the rising bubble and at the interface of the gas and liquid phase. The computational mesh is shown in Figure 1.

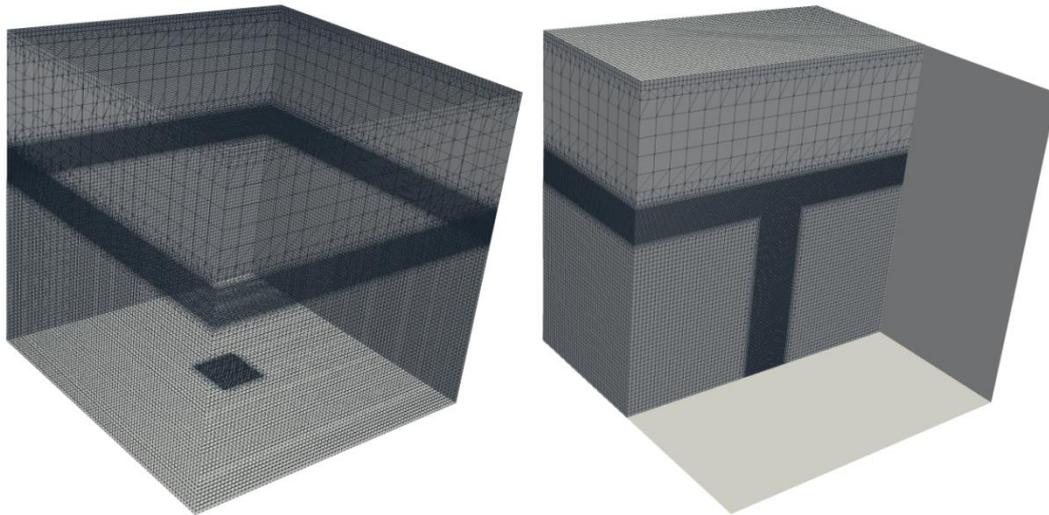


Figure 1. Computational mesh at the surface (left) and meshing inside (right)

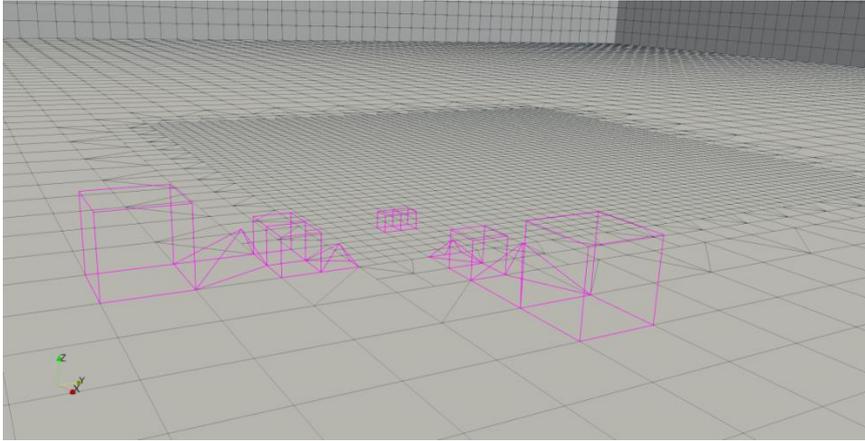


Figure 2. Details of the meshing in the area of the bubble generation

The mesh contains totally 4.1 mill cells, 4.01 mill. cells are hexahedral type, 0.78 mill. cells are polyhedral type. In the area, where the bubble creation and rising is expected, finer mesh resolution was applied in order to detect small changes of the bubble shape. Furthermore, the interaction between liquid and surrounding air (liquid level) is an objective of the investigation as well. Therefore, in this interface area, the fine mesh is adopted as well. Figure 2 shows the mesh surface at the bottom of the container with specific elements. In the layer representing the change of the cell size, different cell types are generated (hexahedral or polyhedral type). This mesh layer influences significantly the mesh quality e.g. the cell volume is 1×10^{-6} for bigger hexahedral cell type, for smaller polyhedral cell type, this volume is of 2.44×10^{-10} . The maximal non-orthogonality is 25.23 and maximal skewness of cells is 0.33. Both values are in tolerance recommended for finite volume methods.

2.4. Foaming Process

The experiment was carried out in the glass storage with dimensions of $30 \text{ cm} \times 20 \text{ cm} \times 20 \text{ cm}$ (12 liters). At the bottom there is a nozzle connected to a flow meter and an air compressor. The surrounding temperature is 299K and air flow rate is setup to be 100L/h. In fact, experiments are performed at different molar ethanol fractions from 0 to 1; in order to illustrate how density and viscosity does change with molar fraction. The experiment started with 1 ethanol molar fraction (X_e) and then water was being added gradually. In order to define changes, values of density and viscosity are measured simultaneously.



Figure 3. Bubbles accumulation after 4 minutes foaming at flow rate 100l/h and $X_e=0.224$

Figure 3 shows typical foam created in the water-ethanol mixture with the molar ethanol fraction of $X_e=0.224$. Experimentally, it was proved that foams are created only in the specific concentration of the Ethanol in the water leading to the specific material property mixture. Furthermore, the Eötvös number was determined to be in order of one.

3. RESULTS

At the bottom of the container, the inlet air jet with the laminar velocity profile is prescribed. The maximal intensity of the inlet air velocity is 0.4 m/s, which correspond to the approx. 5l/h.

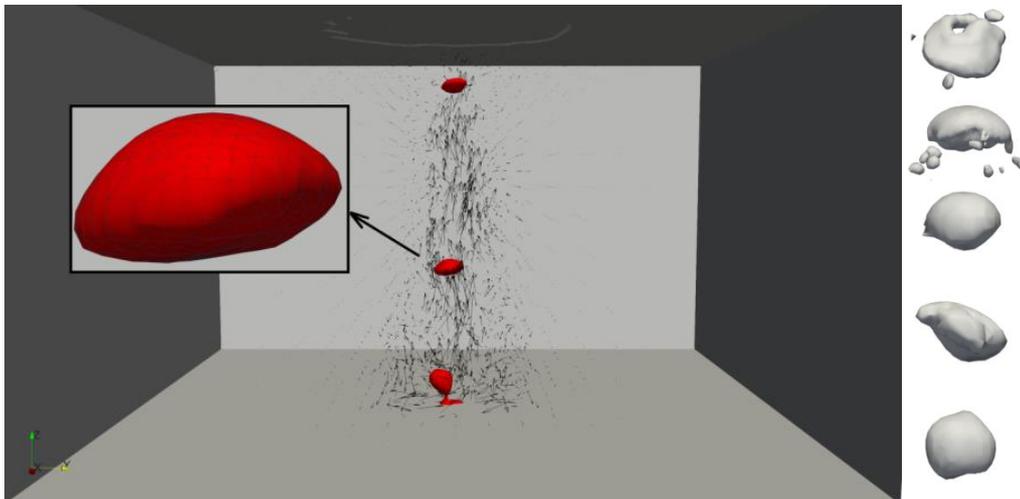


Figure 4. Instantaneous view on formed bubbles under liquid surface level (left) and detailed view on the bubble shapes (right)

This specific flow rate is relatively small which provides flow conditions to observe each bubble separately without merging or other form of the interactions. Figure 4 depicts the rising of bubbles with associated surrounding velocity field. The main flow inside of the container is connected with the dynamics of bubbles. According to e.g. [8], the three-dimensional numerical

study identified an existence of different region of distinct bubble shapes depending e.g. on the Galilei and Eötvös numbers. In case illustrated in Fig. 4, the Galilei number varies between 400 and 800 in dependence on the distance from the gas jet onset. The Eötvös number is in a range between 2,2 to 5,4. For these values the so called IV region associated with spherical cap shape and breakup of bubbles [8]. Numerically, it was observed that how bubbles are rising up, they break up into the spherical caps and later, several smaller satellite bubbles are formed (see Fig. 4 right). Simultaneously, higher values of Galilei number indicate that the buoyancy effect of the bubble will be intensive than an effect of the viscosity. In [10], the relation between the Reynolds number and Weber and Morton numbers takes a form as in Eq. (16).

$$Re = 2.05 We^{2/3} Mo^{-1/5} \quad (16)$$

In the numerical simulation carried out under following conditions defined by the Morton number of order $O(10^{-10})$, the Eötvös number in order of $O(1)$, Weber number in order of $O(1)$ and Galilei number $O(100)$ was found that the bubble Reynolds number is approx. 500. This value of the bubble Reynolds number is in a relatively good match to the bubble Reynolds number calculated using formula in Eq. (16) to be approx. 600. In fact, this kind of the results validation is based on the velocity determination which is much more complicated than bubble diameter analysis for a Ga and Eo numbers. The same conclusion was found in [9] as well. Besides the bubble formation and further propagation, the main objective of the study is concerned on the foam creation.

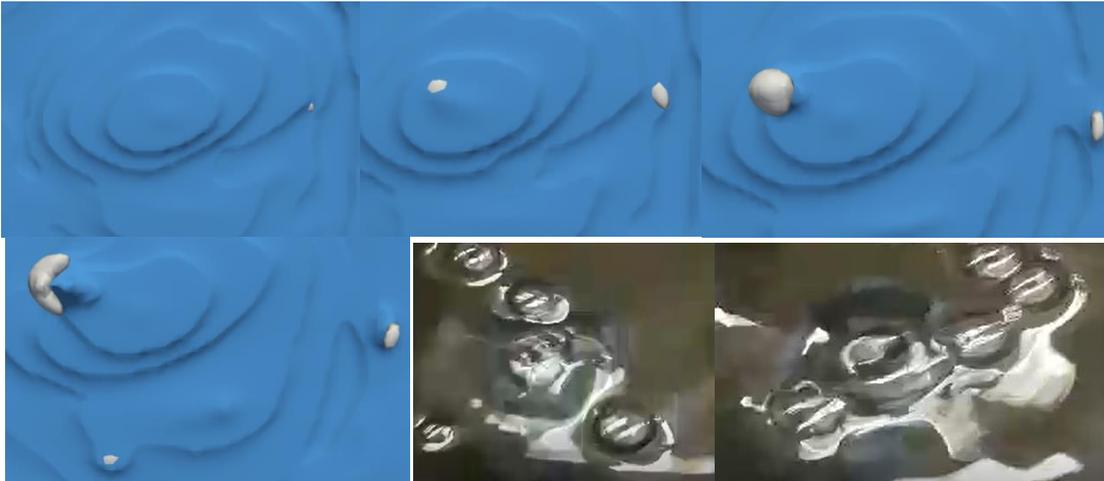


Figure 5. Calculated bubbles every 0.02s captured at the liquid surface level (liquid surface level depicted by blue colour) vs. experiments

Figure 5 depicts the series of bubble formations on the liquid surface with the time step of 0.02s and for lower intensity of foaming process. Bubbles after they reached the liquid surface start to move at the liquid surface towards wall or periodic conditions depending model configuration. The last two pictures in Fig. 5 shows experimental observation of bubbles formed at the liquid surface. Similarly to numerical results, bubbles are moving separately in same direction, however, no bubble break up is observed in contrast to numerics.

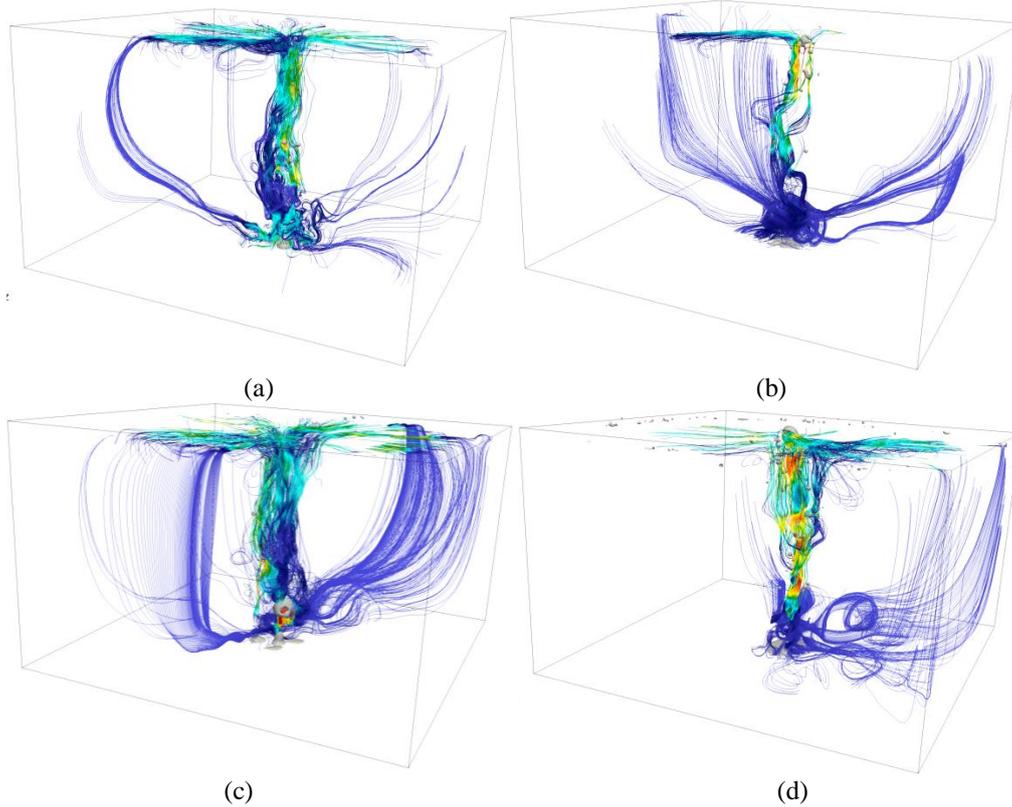


Figure 6. Path lines with velocity depictions for periodic boundaries at the vertical boundary (a) in time 2.91s and (b) 3.45s and for walls at the vertical boundary (c) 2.91s and (d) 3.45s

Therefore, the detailed study is required to simulate bubble behaviour just at the liquid surface, where interface between gas and surrounding liquid phase and again gas phase is defined by very small thickness. This fact can be reason that bubbles cannot keep form and start to break up or dissolve. This so called soap film behaviour differences phases can be expressed by the balance between forces on the film in terms of the pressure difference [7] given by Eq. (17).

$$\Delta p = \frac{8\sigma}{D} \quad (17)$$

The foaming process and resulting foams are influenced by the side walls. Figure 6 shows path lines in the investigated domain for periodic or wall side boundary conditions. Colours indicate the intensity of the velocity, where red colour corresponds to the 0.4 m/s which is maximal velocity in the core of the air jet used in the air injection at the bottom. The flow at the liquid surface level is responsible for the bubble transport from the centre of container towards side wall. The liquid surface velocity is asymmetrical any boundary conditions and varies on the time. In case of wall conditions, the entire domain is influenced by gas jet, however, in case of periodic conditions; the upper part of the container is less influenced by the gas jet flow.

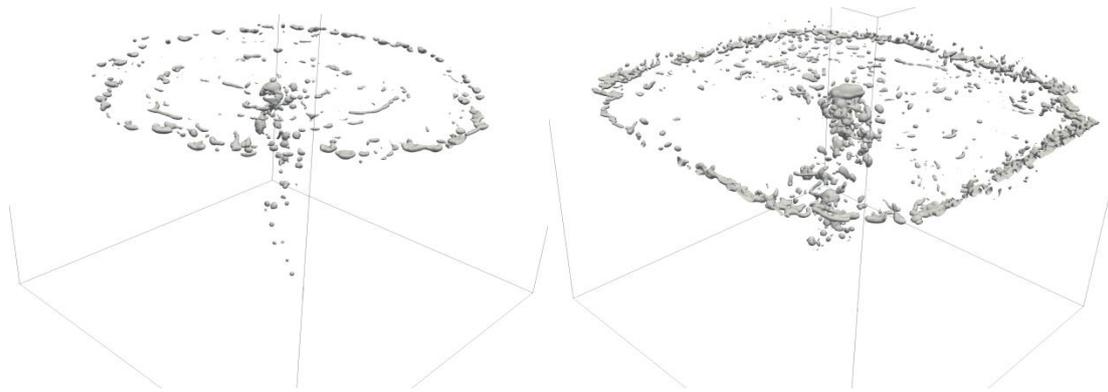


Figure 7. Bubble structures at the liquid free surface for periodic side vertical boundary (left) and wall type vertical boundary (right)

Figure 7 shows the air structures of bubbles which are rising upwards and simultaneously bubble structures captured at the free surface of the liquid for periodic or wall type vertical boundaries. It is evident that higher accumulation of the gas phase is concentrated if the wall type conditions are prescribed at the vertical walls. To achieve so high concentration of bubbles at the liquid free surface, a long time simulation is required particularly more than 2s of calculation. The time step of the calculation is limited to the maximal courant number of 0.95 however, in fact, this value is varying around 0.6. The size of the time steps depends on the maximal velocity appeared in the velocity field. It was detected in simulations the significant velocity oscillation mostly by the insipience of the bubbles at the bottom. These higher velocities reduce for a particular time period the size of the time step used for calculations. In fact the time step is varying from 5×10^{-5} s to 2×10^{-4} s.

4. CONCLUSIONS

The two-phase flow problem of the bubble formation on the created foam at the liquid free surface was studied numerically using the compressible multi-phase flow solver based on the Volume of Fluids approach. For fluid phase, the mixture of ethanol-water with specific molar fraction of the ethanol was considered to replace real aluminum alloys. The temperature effect and solidification are completely neglected. The main objective of the study is the bubble behavior in this isothermal homogeneous liquid mixture and the foaming process. It was found that the compressible multi-phase flow solver was able to predict the bubble formation and its dynamics. Results were compared with other experimental and numerical results. Furthermore, using appropriate structure visualization, the original bubbles are possible to still observe later on the liquid free surface where they stay to be floating. However, the strong gas phase dissipation appeared maybe because of the insufficient mesh resolution. This part of the flow at the free surface level needs to be investigating further. On the other hand, different requirements for the mesh resolution causes non-homogenous mesh in which the mesh quality must be carefully check it. In perspectives, the simulation will continue to find main differences between mixture of water-ethanol and poor water. Simultaneously, the impact of the side boundaries on the foams will be going on. In this study, it was found that vertical boundaries can influence the foaming process and in case of the walls at the vertical boundaries, higher concentration of gas phase fraction was observed.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic and the European Union - European Structural and Investment Funds in the frames of Operational Programme Research Development and Education - project Hybrid Materials for Hierarchical Structures (HyHi, Reg. No. CZ.02.1.01/0.0/16-019/0000843).

REFERENCES

- [1] Sebastian Aland, Stephan Schwarz, Jochen Fröhlich & Axel Voigt, (2013) “Modeling and numerical approximations for bubbles in liquid metal“, *Eur. Phys. J. Special Topics* 220, 185 –194.
- [2] Karel Fraňa, Shehab Attia, Jörg Stiller, (2019) “A Bubble Formation in the Two-Phase System“, In: Rodrigues J. et al. (eds) *Computational Science – ICCS 2019. Lecture Notes in Computer Science*, vol 11539. Springer.
- [3] C.W. Hirt & B.D. Nichols, (1981) “Volume of fluid (VOF) method for the dynamics of free boundaries“, *J. Comput. Phys.* 9(1), 221-225.
- [4] Karel Fraňa & Iva Nová, (2018) “A interface formation in two-phase flow problems“, *International Journal of Mechanics*, 12, pp. 102-108.
- [5] Y. Yang, S. Wang & C. Wen, (2017) “Gas-liquid two-phase flows in double inlet cyclones for natural gas separation“, *Cogent Engineering* 4(1).
- [6] A. Hasečić, S. Muzaferija & I. Demirdžić, (2020) “Finite volume method for multiphase flows with radiation and phase change“, *International Journal of Thermal Sciences*, Vol. 149, 106201.
- [7] E. Rio & AL. Biance, (2014) “Thermodynamic and mechanical timescales involved in foam film rupture and liquid foam coalescence“, *Chemphyschem*, Dec;15(17), pp. 3692-3707. doi:10.1002/cphc.201402195.
- [8] D.M. Sharaf, A.R. Premlata, Manoj Kumar Tripathi, Dadarinath Karri & Kirti Chandra Sahu, (2018) “Shapes and paths of an air bubble rising in quiescent liquids“, *Physics of Fluids* 29, 122104, <https://doi.org/10.1063/1.5006726>.
- [9] Dominique Legendre, Roberto Zenit, & J. Rodrigo Velez-Cordero, (2012) “On the deformation of gas bubbles in liquids“, *Physics of Fluids* 24, 043303, <https://doi.org/10.1063/1.4705527>.
- [10] Marie Rastello, Jean-Louis Marié & Michel Lance, (2010). “Drag and lift forces on clean spherical air bubbles in a rotating flow“, *J. Fluid Mech.* 682, 434.
- [11] Mitsuhiro Ohta & Mark Sussman, (2012) „The buoyancy-driven motion of a single skirted bubble or drop rising through a viscous liquid“, *Physics of Fluids* 24, 112101, <https://doi.org/10.1063/1.4765669>.
- [12] M.K. Tripathi, K.C. Sahu & R. Govindarajan, (2015) “Dynamics of an initially spherical bubble rising in quiescent liquid” *Nature communications*, 6, 6268.

AUTHORS

Karel Frana Professor at the Technical university of Liberec, Faculty of Mechanical Engineering, research field: numerical simulations, multiphase flow problems, energy efficiency; More than 20 publications in International Journals, more than 50 conference papers, more than 10 contributions in books.



Jörg Stiller Professor at Technische Universität Dresden, Institut für Strömungsmechanik Research field: computational fluid dynamics, multi-level methods with higher orders, turbulent flows, aerodynamic Recently: more than 60 papers in journals or conferences.



Iva Nová Senior Professor at the Technical university of Liberec, Faculty of Mechanical Engineering, research field: metal structures, foaming processes, experimental investigation, metal solidifications; More than 20 publications in International Journals, more than 50 conference papers, more than 10 contributions in books.



DOCPRO: A FRAMEWORK FOR BUILDING DOCUMENT PROCESSING SYSTEMS

Ming-Jen Huang, Chun-Fang Huang, Chiching Wei

Foxit Software Inc., Albrae Street, Fremont, USA

ABSTRACT

With the recent advance of the deep neural network, we observe new applications of natural language processing (NLP) and computer vision (CV) technologies. Especially, when applying them to document processing, NLP and CV tasks are usually treated individually in research work and open source libraries. However, designing a real-world document processing system needs to weave NLP and CV tasks and their generated information together. There is a need to have a unified approach for processing documents containing textual and graphical elements with rich formats, diverse layout arrangement, and distinct semantics. This paper introduces a framework to fulfil this need. The framework includes a representation model definition for holding the generated information and specifications defining the coordination between the NLP and CV tasks.

KEYWORDS

Document Processing, Framework, Formal definition, Machine Learning.

1. INTRODUCTION

Business documents nowadays are usually composed of multiple types of information, such as text, images, tables, charts, formulas. Their semantics, formats, and styles are also abundant. To create a system to assist humans in reading, comprehension, and writing documents, there is a need to combine various technologies for analyzing textual and graphical elements. In addition to this, the analyzed results must be able to be stored and consumed by machines.

Document processing has been long considered an application of Natural Language Processing (NLP) [1], such as named entity recognition, sentiment analysis, semantic relations. Another application is to apply Computer Vision (CV) for document-layout analysis [2], which is to determine document structure by detecting locations, bounding boundary, and types of document elements. Another prominent CV task related to document processing is image captioning [3]. All of these individual NLP and CV tasks are already quite common in academic research. Open source components are also developed for many years [4, 5, 6].

To fulfill the requirements for building such a system, we propose an architecture design as a blueprint for building a system that can process documents with rich formats, styles, and multiple types of elements. The architecture includes (1) a document representation model definition that can be instantiated with analyzed data and can be consumed by other software components and (2) a customizable framework that coordinates various tasks for analyzing documents. We define the framework with formal definitions and illustrate with examples.

In this paper, we firstly describe previous NLP and CV research work. We then describe the overall architecture of the framework. Finally, we detail the document representation model and the task coordination definition.

2. PREVIOUS WORK

Document processing is a vast area with many topics. We listed some of the topics below.

NLP has been applying to various document processing tasks. Name entity recognition (NER) is a task of identifying the type of an entity within the text. Supervised learning approaches usually required to prepare a dictionary or annotated datasets [7, 8]. Even this approach can create a high-performance NER model, it is a time-consuming task and needs multi-language annotated datasets. On the other hand, the weakly supervised approach starts entity classification with a small dataset or rules and expanding more rules with new iterations [9].

Similar text analysis is a task to detect similarity between sentences, paragraphs, and documents. One of the most common approaches is to calculate various types of distances between text vector spaces [10]. The vector spaces could be calculated from terms, corpus, or knowledge [11, 12, 13].

Text classification is a task to assign a category to a document. Classification approaches are diverse. Among others, SVM (support vector machine) demonstrates that it is an efficient approach for document classification [14]. More recent research applies deep learning, such as CNN [15] and CNN-LSTM [16], for document classification. Hingmire et al. [17] chain two NLP tasks. They first apply topic modeling and text classification. This approach can provide a categorization explanation and high accuracy at the same time.

Summarization is a task to create a shorter version of documents with primary ideas. There are two types of output, abstractive and extractive. The abstractive summary is to generate new sentences that are in the original documents. The extractive summary, regarded as a problem of classification, is composed of sentences or paragraphs in the original documents. In general, the extractive summary can be considered as a classification problem, that is, whether a sentence is a summary sentence or not. Various approaches are proposed [18, 19]. More recent research work applies deep learning [20, 21].

CV is applied to solve image-based information of documents. Document layout analysis detects objects and classifies them into different categories. Recent work usually applies CNN for analyzing document layout. Julca-Aguilar et al. propose CNN for detecting text/non-text document elements [22].

Image captioning is a task to give a natural language description to an image. It is a relatively new research area where the chaining of CV and NLP tasks becomes prevalent [23]. Two common approaches are (1) capturing the main point of an image and generating a description for it [24] and (2) generating a description of each detected object and combined the descriptions [25]. Anderson et al. [26] combine both (1) and (2) approaches to provide different levels of details. Grounded language acquisition is a representatively interdisciplinary field of CV and NLP integration [27]. It requires both disciplines to map language representation to real-world objects. Mavridis and Roy [28] define an architecture with language understanding, visual perception, and action modules for robots and human cooperation tasks.

Truica et al. [29] propose a data processing framework with a flexible data model with several preprocessing techniques. Dawborn, T., & Curran, J. R. [30] propose and implement a document model for document representation. It is relatively rare to work from a comprehensive approach for document processing.

In brief, to apply various NLP and CV tasks to real-world business scenarios, a unified framework is necessary.

3. ARCHITECTURE

3.1. Overall Design

Figure 1 shows a conceptual diagram of the framework, which is called DocPro. The basic building blocks of DocPro include several tasks for processing input documents and generating target documents. The processing of documents creates one or more document representation models that are designed for storing analyzed results.

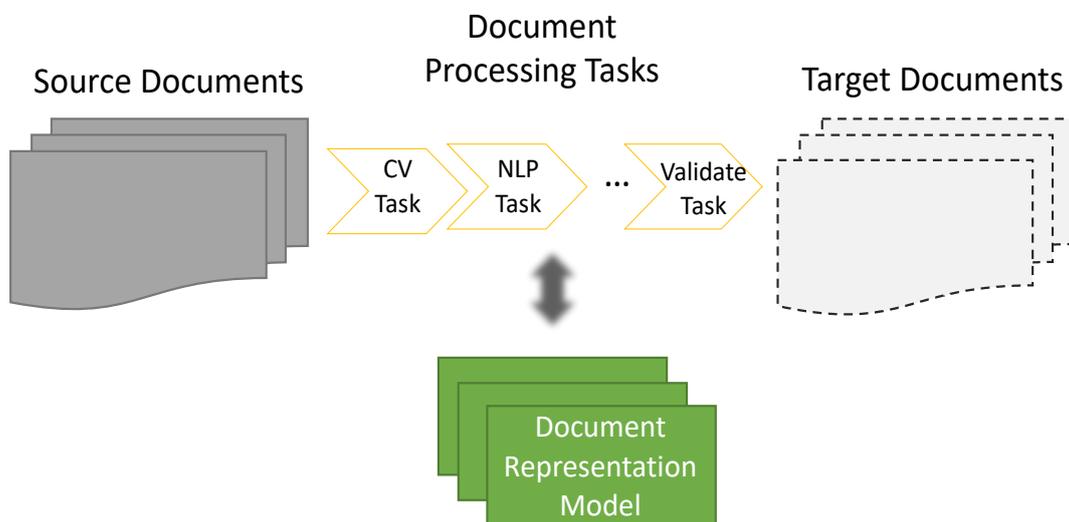


Figure 1. Conceptual Diagram of the Document Processing Framework (DocPro)

We can assign multiple tasks based on business objectives. There are four types of tasks:

- Tasks to analyze the layout of documents
- Tasks to understand the textual elements of the documents
- Tasks to understand the graphic elements of the documents
- Tasks to write a document

3.2. Document Representation Model

The paper defines a document representation model to accommodate all of the necessary information for the defined tasks with a structure for holding the following information:

- Document layout. It includes the locations and boundaries of each document element. The types of document elements are diverse, such as footers, headers, paragraphs, charts, tables, images, formulas, and paragraphs.

- Document reading order. The semantic order of document objects of a document
- Document summary. A paragraph to describe the critical meaning of the original documents.
- Document metadata. Information extracted from documents, such as dates, time, person names, location names, organization names, and as such.
- Document text. All of the textual data of a document originated from various document elements, such as paragraphs, image captions, table captions.
- Document graphical description. Textual description of graphical elements. Data can be from image and chart captions already written in a document or decoded from image pixels by machines.

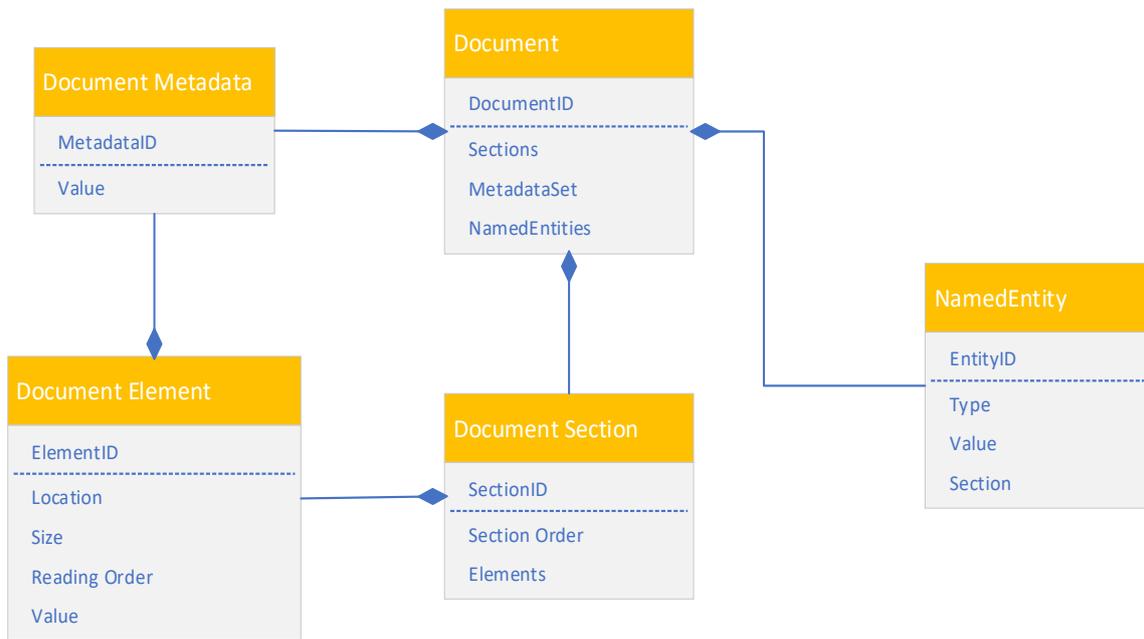


Figure 2. The Core Model Definition of DocPro

The definition of the model consists of two parts. The first is the Core Model definition, a lightweight and concise, and serves as the base model for extending. Figure 2 shows its definition. The central class is the Document entity, which represents a single document. The Document Metadata entity stores a piece of meta-information of a document. The NamedEntity represents all detected entity names, such as person names, location names, and organization names. The Document Section is a semantic segment for holding document elements. The Document Element is an object of a document like an image or a chart.

We can define a new model definition based on the Core Model definition to support other specific scenarios. Figure 3 shows an extended model definition. In this extended version, there are several entities extended from the core model. The Label and Category entities are extended from the Document Metadata entity for holding the labels and category information of a document. The Image, Formula, Header, Footer, Chart, and Table entities are all extended from the Document Element entity. The Page entity is extended from the Document Section entity, which represents a page of a document, which also holds many different Document Element entities like images, formula, header, footer, charts, and tables. From the above description, we can conclude that this extended model definition could represent a book or an academic paper. Both layout information and semantics information of books and articles are defined.

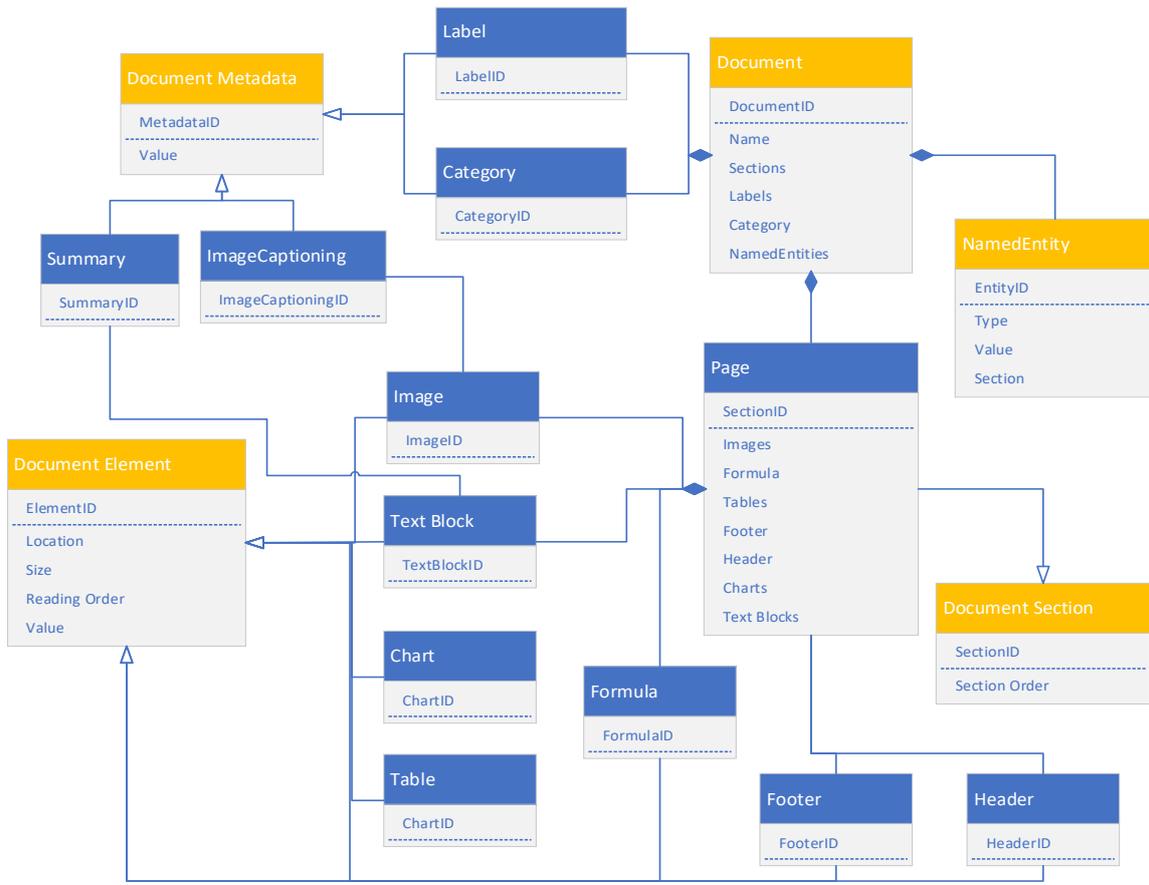


Figure 3. The Extension Model Definition for Standalone Document

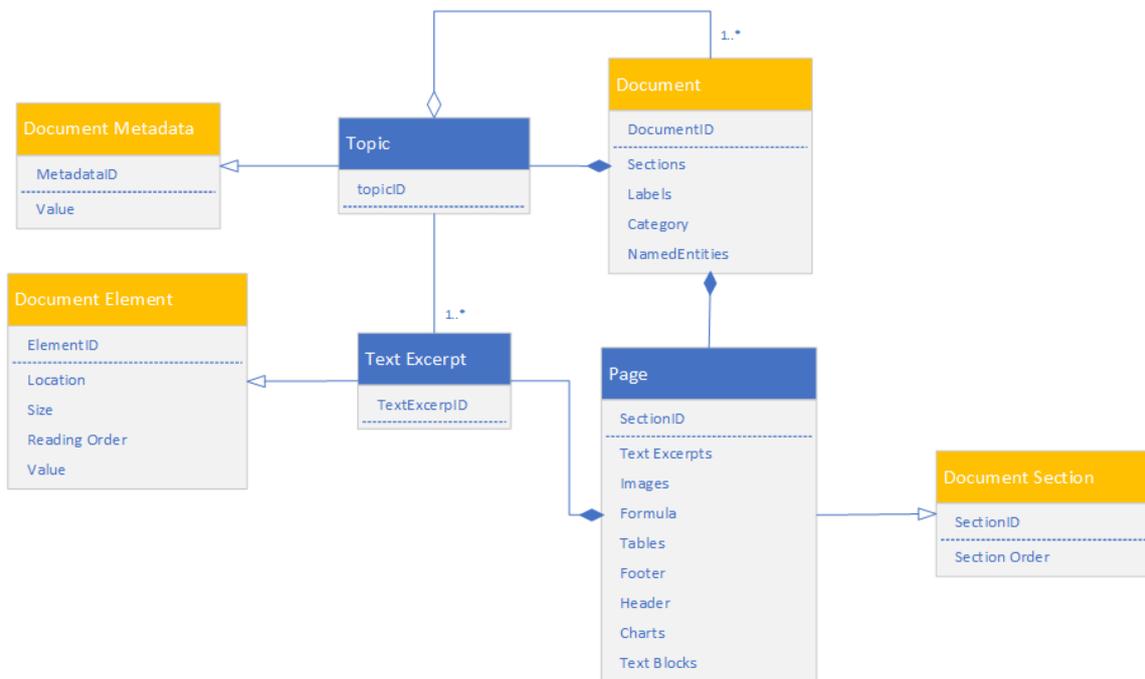


Figure 4. The Extension Model Definition for Cross-Document Knowledge Correlation

Figure 4 is an Extension Model Definition for describing several documents correlated by some topics. A document could have one or more topics, where each topic represented by a Topic class. A topic is associated with some text excerpts, defined as the Text Excerpt entity. A topic associated with one or more documents models the correlation among documents. With this definition, we can model a knowledge map with many topics discovered from multiple documents.

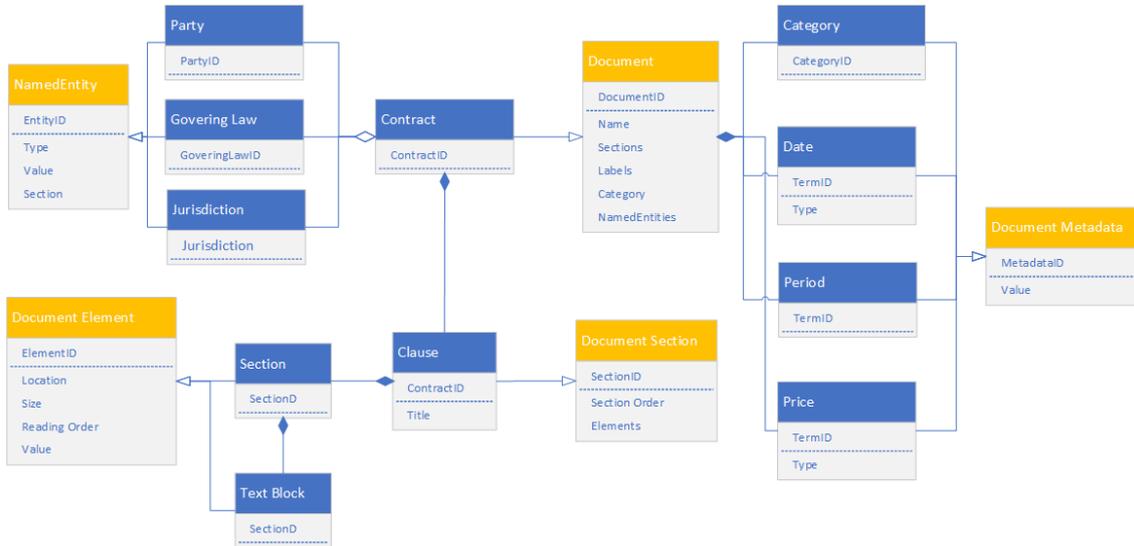


Figure 5. The Extension Model Definition for Contracts

Figure 5 is another Extension Model definition defined for a more specific domain – contract review. Contract tasks are essential activities of any business. The tasks might include contract monitoring, reviewing, and drafting [29]. This model is also defined based on the Core Model shown in Figure 2. The central component becomes the Contract entity inherited from the Document entity. The Clause entity is to model contract articles and clauses. A clause consists of one or more sections, which are represented by the Section entity. The Text Block entity models the textual description of a section. Special items, such as recitals and preambles, can also be described by the Clause and section entities.

Temporal information is modeled by the Date class, which can hold contract start, termination, effective dates. The Value entity contains financial information. The Period entity represents the number of days a contract is valid. Inheriting from the NamedEntity entity represents other important information like contracting parties, governing law, and jurisdiction.

We can observe from the description above that the document representation model is beneficial for many business scenarios. One exemplified scenario is to create a tool for editing scanned documents with the information above. Another scenario can be business activity automation, like contract monitoring and review. With such core and extension model definitions, machines could streamline the decision-making process based on document contents.

4. DOCUMENT PROCESSING COORDINATION FRAMEWORK

In this section, we describe a framework for coordinating various types of tasks for analyzing documents with formal definitions and examples.

The limitation of current document processing technologies is isolated tasks as designed. However, with the rich format, styles, and semantics of documents, the integral of document analysis work becomes vital to anyone who wants to create any document processing system. Another critical factor in designing the framework is that the types of tasks do not limit to machine learning implementation. We do not consider that all of the tasks for solving document processing problems would belong to the machine learning type. Therefore, the logic within each task is varied, including tasks like machine learning type for predictive work, deterministic computation type for analysis, and rule-based type for reactive responses.

The framework defines the five types of elements:

- Data source
- Tasks
- Model
- Checker
- Process

A task consumes one or more documents emitted from a data source. A checker is a task to validate and verify the current status of models. A model is an instance for representing the analysis results of a document representation, defined in the previous section. A process is a series of connected tasks and checkers.

The framework does not specify formats of source documents. They can be PDF, MS Word, images, or others. The logic of tasks can be various, such as machine learning, rule-based, and even simple deterministic procedure. Each task should provide new information with more details and might create a new model or update existing a model with more accurate information.

Checkers are tasks designed explicitly for validating models. Checkers are attached to a model or can be attached to elements of a model for checking the whole model or parts of the model.

Tasks and checkers can be chained together to react to the update of a model. For example, after an OCR task updates a model, a checker is triggered to check if adding new paragraphs and executing a task to classify the document into a category. Another checker checks if a document category is an employment contract and executes a new task for automatically reviewing contract contents by pre-defined rules. A process manages these chained tasks, checkers, documents, and models.

The behavior of a checker is a tuple $C = (M, E, L, T)$

- M is a set of document representation models.
- E is a set of elements of a document representation model m .
- L is a set of checker labels.
- T is a set of conditional tasks.

The behavior of a process is a tuple $P = (T, D, M, C)$

- T is a set of tasks.
- D is a set of source documents.
- M is a set of document representation models.
- C is a set of checkers.

An M consists of one or more entities defined in Section 3. It always includes a Document entity with several associated objects. For example, after applying a document-layout analysis task to a document, an m_o is initiated with a Document entity and a set of associated objects, including a Header, a Footer, and many Text Blocks. These associated objects are E defined above.

T of P above can further be expressed as $p^{i \rightarrow j}$ is starting from executing task t_i and ending at t_j . m_i is the model updated by t_i and m_j is updated by t_j . The process can also be denoted as follows:

$$p_{i \rightarrow j} = t_i(D) \rightarrow t_{i+1}(m_i) \rightarrow t_{i+2}(m_{i+1}) \dots \rightarrow t_j(m_{j-1}),$$

where $\forall n = \{i, j\}: (t_n, m_n, t_{n+1})$

We can also add checkers at the end of the process and denote the process as follows:

$$p_{i \rightarrow j} = t_i(D) \rightarrow t_{i+1}(m_i) \rightarrow t_{i+2}(m_{i+1}) \dots \rightarrow t_{j-1}(m_{j-2}) \rightarrow c_j(m_{j-1}),$$

where $\forall n = \{i, j\}: (t_n, m_n, t_{n+1}) (t_{n+1}, m_{n+1}, c_{n+2})$

Below we use eight examples to illustrate the syntax and semantics of this formal definition.

Example 1 – OCR

d_0 is a scanned academic paper, t_a is a task to recognize text elements from d_0 , m is a model for storing the recognized results:

$$t_a = (d_0, ocr, m_0)$$

Example 2 – Summarization

t_b is a task to analyze text blocks for generating summaries. When it is applied to the same document d_0 as Example 1:

$$t_b = (d_0, summarize, m_0)$$

Example 3 -Topic Discovery

t_c is a task for discovering topics from d_0 . However, this task relies on parsed textual elements. We need a conditional task c_{t_e} with a checker.

$$t_c = (d_0, \text{topic} - \text{discovering}, m_0)$$

$$c_{te} = (m_0, e_{txt}, \text{check} - \text{textual} - \text{elements}, t_c)$$

Example 4 – Topic Correlation

t_d is a task to discover related topics from documents d_r for a target document d_0 . In this example, t_d is applied to a model the same as to Examples 1 ~ 3. A checker c_{toc} is defined for checking topics discovered from documents.

$$t_d = (d_r, \text{correlate} - \text{knowledge}, m_0)$$

$$c_{toc} = (\{m_0, m_r\}, e_{toc}, \text{check} - \text{topics}, t_d)$$

Example 5 – Knowledge Mapping Automation

A process p_0 of building a knowledge map with connected documents correlated by topics. The process includes the tasks and checker of Examples 1 ~ 4. A process is usually for defining a data processing pipeline.

$$p_0 = t_a(d_0) \rightarrow t_b(m_0) \rightarrow c_{te}(m_0) \rightarrow c_{toc}(m_0)$$

Example 6 – Report Generation

t_r is a task to fine-tune a language model LR for generating financial reports. D_t is a set of financial documents to be used as training data for t_r . t_w is a task for generating a report R from a specific topic. A checker c_{rpt} is defined for checking topics discovered from documents and execute t_w .

$$t_r = (D_t, \text{fine} - \text{tune}, LR)$$

$$t_w = (M, \text{write}, R)$$

$$c_{rpt} = (M, e_{toc}, \text{check} - \text{topics}, t_w)$$

Example 7 – Contract Review Assist

d_c is a contract. t_e is a task to classify a contract into a category. t_r is a task to extract essential data, such as contractual party names, start/end dates, and monetary values, in d_c . m_c is a model for storing contractual elements. t_l is a task to analyze the contract layout. More specifically, it recognizes clauses appearing in d_c . c_{cpvl} is a checker checking the contract category, parties, dates, values, and layout of the contract in m_c . t_f is a task to forward the contract to the right person for further review based on the analyzed results by t_e , t_r , and t_l .

$$t_e = (d_c, \text{classify}, m_c)$$

$$t_r = (d_c, \text{recognize} - \text{named} - \text{entity}, m_c)$$

$$t_l = (d_c, \text{analyze} - \text{layout}, m_c)$$

$$t_f = (d_c, forward, m_c)$$

$$c_{cpv} = (m_c, \{e_{category}, e_{parties}, e_{value}, e_{layout}\}, check - contract - errors, t_f)$$

Example 8 – Contract Review Automation

The first requirement of the contract automation system is to automate the process of contract routing. Most of the components are already defined in Example 7. The process p_c is defined as below:

$$p_c = t_e(d_c) \rightarrow t_r(m_c) \rightarrow t_l(d_c) \rightarrow c_{cpvl}(m_c)$$

6. CONCLUSIONS AND FUTURE WORK

The primary contribution of this paper is two folds. First, we proposed a document representation model that has a lightweight and concise core definition. The core definition can extend to define more document types used in different business scenarios. Secondly, formal definitions of a document processing framework are detailed. A modern document processing application integrating with various types of tasks can be built based on the framework. The format definitions can become the design language of any such systems.

Our future work is to create an open-source project, including the model definition, software components, and API definitions used for creating real-world systems for processing documents in different business scenarios.

REFERENCES

- [1] Brants, T. (2003, September). Natural Language Processing in Information Retrieval. In CLIN.
- [2] Breuel, T. M. (2003, April). High performance document layout analysis. In Proceedings of the Symposium on Document Image Understanding Technology (pp. 209-218).
- [3] Liu, X., Xu, Q., & Wang, N. (2019). A survey on deep neural network-based image captioning. The Visual Computer, 35(3), 445-470.
- [4] OpenCV official web site. (<https://opencv.org/>)
- [5] Gensim official web site (<https://radimrehurek.com/gensim/index.html>)
- [6] NLP Architect by Intel (http://nlp_architect.nervanasys.com/)
- [7] GuoDong, Z., & Jian, S. (2004, August). Exploring deep knowledge resources in biomedical name recognition. In Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (pp. 96-99). Association for Computational Linguistics.
- [8] Jiang, R., Banchs, R. E., & Li, H. (2016, August). Evaluating and combining name entity recognition systems. In Proceedings of the Sixth Named Entity Workshop (pp. 21-27).
- [9] Irmak, U., & Kraft, R. (2010, April). A scalable machine-learning approach for semi-structured named entity recognition. In Proceedings of the 19th international conference on World wide web (pp. 461-470).
- [10] Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).
- [11] Huang, A. (2008, April). Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand (Vol. 4, pp. 9-56).
- [12] Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, 104(2), 211.
- [13] Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).
- [14] Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. Journal of machine Learning research, 2(Dec), 139-154.

- [15] Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058.
- [16] Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630.
- [17] Hingmire, S., Chougule, S., Palshikar, G. K., & Chakraborti, S. (2013, July). Document classification by topic labeling. In Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval (pp. 877-880).
- [18] Wang, L., & Cardie, C. (2013, August). Domain-independent abstract generation for focused meeting summarization. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1395-1405).
- [19] Liao, K., Lebanoff, L., & Liu, F. (2018). Abstract meaning representation for multi-document summarization. arXiv preprint arXiv:1806.05655.
- [20] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.
- [21] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.
- [22] Julca-Aguilar, F. D., Maia, A. L., & Hirata, N. S. (2017, October). Text/non-text classification of connected components in document images. In 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 450-455). IEEE.
- [23] You, Q., Jin, H., Wang, Z., Fang, C., & Luo, J. (2016). Image captioning with semantic attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4651-4659).
- [24] Chen, X., & Zitnick, C. L. (2014). Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654.
- [25] Elliott, D., & Keller, F. (2013, October). Image description using visual dependency representations. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1292-1302).
- [26] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6077-6086).
- [27] Wiriyathammabhum, P., Summers-Stay, D., Fermüller, C., & Aloimonos, Y. (2016). Computer vision and natural language processing: recent approaches in multimedia and robotics. *ACM Computing Surveys (CSUR)*, 49(4), 1-44.
- [28] Mavridis, N., & Roy, D. (2006, October). Grounded situation models for robots: Where words and percepts meet. In 2006 IEEE/RSJ international conference on intelligent robots and systems (pp. 4690-4697). IEEE.
- [29] Truică, Ciprian-Octavian, Jérôme Darmont, and Julien Velcin. "A scalable document-based architecture for text analysis." *International Conference on Advanced Data Mining and Applications*. Springer, Cham, 2016.
- [30] Dawborn, T., & Curran, J. R. (2014, August). docrep: A lightweight and efficient document representation framework. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 762-771).
- [31] Milosevic, Z., Gibson, S., Lington, P. F., Cole, J., & Kulkarni, S. (2004, July). On design and implementation of a contract monitoring facility. In Proceedings. First IEEE International Workshop on Electronic Contracting, 2004. (pp. 62-70). IEEE.

OBJECT DETECTION IN TRAFFIC SCENARIOS - A COMPARISON OF TRADITIONAL AND DEEP LEARNING APPROACHES

Gopi Krishna Erabati, Nuno Gonçalves and Hélder Araújo

Institute of Systems and Robotics, University of Coimbra, Portugal

ABSTRACT

In the area of computer vision, research on object detection algorithms has grown rapidly as it is the fundamental step for automation, specifically for self-driving vehicles. This work presents a comparison of traditional and deep learning approaches for the task of object detection in traffic scenarios. The handcrafted feature descriptor like Histogram of oriented Gradients (HOG) with a linear Support Vector Machine (SVM) classifier is compared with deep learning approaches like Single Shot Detector (SSD) and You Only Look Once (YOLO), in terms of mean Average Precision (mAP) and processing speed. SSD algorithm is implemented with different backbone architectures like VGG16, MobileNetV2 and ResNeXt50, similarly YOLO algorithm with MobileNetV1 and ResNet50, to compare the performance of the approaches. The training and inference is performed on PASCAL VOC 2007 and 2012 training, and PASCAL VOC 2007 test data respectively. We consider five classes relevant for traffic scenarios, namely, bicycle, bus, car, motorbike and person for the calculation of mAP. Both qualitative and quantitative results are presented for comparison. For the task of object detection, the deep learning approaches outperform the traditional approach both in accuracy and speed. This is achieved at the cost of requiring large amount of data, high computation power and time to train a deep learning approach.

KEYWORDS

Object Detection, Deep Learning, SVM, SSD & YOLO

1. INTRODUCTION

Being inspired from nature or human physiology, the science and technology are progressing rapidly to leverage human sophistication and well being. The tools or technology that is being developed is an extension of human faculties, for instance, as humans have the ability of locomotion, there arose the development of vehicles like cars or bikes for faster locomotion. In the past few decades, the automobile industry, research units and academia is focusing on automation of vehicles (self-driving vehicles). One of the significant reasons for automation of vehicles is to trim down the accidents caused by human drivers. Distraction [1] (due to texting, tiredness etc.), speeding [2], drunk driving [3], recklessness are some of the instances the human driver would create problems in traffic environments. The consequences of these actions may vary from damage to property to loss of life. On an average, every minute at least one person dies in a vehicle accident. Not only loss of life but also on an average 10 million people are injured each year. The hospital costs, damage to property and other costs will sum up to 1-3 % of world's gross domestic product [4].

During this era of automation, scientific and technical community has been providing various solutions to these problems. Pre-crash systems is an active area of research to reduce accident severity and injury. The major threat a driver faces in a traffic scenario is from other vehicles. Advanced Driver Assistance Systems (ADAS) [5] which warn the driver about possible

collision with other vehicles has gained lot of attention in research. Robust, reliable and fast obstacle detection is very significant and fundamental step in such systems.

In traffic scenarios, obstacle/object detection is the most important task for autonomous driving, since the relevant scene understanding is essential for driving controls. In computer vision literature, object detection is a twofold process, to classify which category object belongs to (object classification) and to determine the location of object in the scene (object localization). In traffic scenarios, objects include cars, bikes, trucks, buses, pedestrians etc. The object detection algorithms take input in the form of 2D data (like RGB images) and provides the output in the form of 2D bounding boxes which localizes the object along with the class of detected object.

The traditional object detection algorithms extract handcrafted features (like SIFT, SURF etc.) for semantic representation and use shallow networks or statistical pattern recognition to classify the object. On the other hand, deep learning [6] networks learn more complex features and semantic object representations which are leveraged by classifier networks.

This paper presents a comparison of traditional and deep learning approaches for object detection in traffic scenarios. A traditional object detection pipeline with HOG [10] features and SVM [13] classifier is compared with more advanced deep learning object detection approaches like SSD [33] and YOLO [32]. The SSD network is trained and tested with different backbone networks, such as VGG16 [20], MobileNetV2 [8] and ResNeXt50 [9], similarly YOLO with MobileNetV1 [39] and ResNet50 [21]. The three approaches are trained with PASCAL VOC 2007 and 2012 [7] dataset and tested with PASCAL VOC 2007 test set. Both qualitative and quantitative results are presented for comparison. The mAP for five classes, namely, bicycle, bus, car, motorbike and person is calculated for three approaches along with the processing speed.

This paper has the following structure: related work of object detection is presented in Section 2, an overview of compared methods is discussed in Section 3, training details are presented in Section 4 followed by results and discussion in Section 5, and finally conclusion in Section 6.

2. RELATED WORK

The topic of object detection has been widely researched in the last two decades. The aim of object detection algorithms using 2D data (RGB images) is to classify and locate the objects in the images, with 2D rectangular bounding boxes and also show their confidence of existence. This task is performed by the handcrafted traditional methods and new age deep learning methods. Depending on the methods to extract semantic features from the images to classify the objects, we divide the object detection methods into two: traditional methods and deep learning methods. The workflow of the two methods is shown in Figure 1.

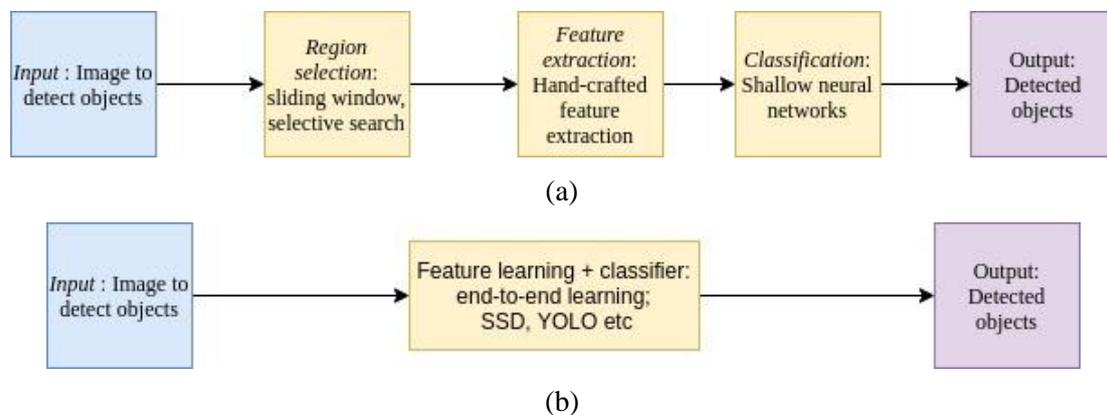


Figure 1. Workflow of (a) traditional methods and (b) deep learning methods

Features are compact, yet descriptive and distinctive attributes of the objects in the images, which are classified with classifiers. In traditional methods the features are handcrafted using established computer vision feature descriptors like SIFT [11], SURF [12] etc. The features are extracted as many as possible practically and form a definition known as bag of words for each object class. But there is always a difficulty to choose the priority of features to be extracted from the given data. So, it is up to the designer's judgement which features would best describe the different classes of objects in the data. The features are passed on to a shallow network classifier for classification of objects. In deep learning methods, the features are extracted by the network from the data, by discovering underlying patterns in the classes of data and automatically working best to provide compact, yet salient features for each class of object.

2.1 Traditional methods

The traditional object detection pipeline can be divided into three stages: Region selection, Feature description and Classification. *Region selection*: The different objects which are to be detected may appear at any position in the image and have variety of sizes or aspect ratios. So, it is a natural choice to search the entire image with multi-scale sliding windows for the region selection. This approach may result in finding all possible positions of objects, but it is very time consuming and computationally expensive due to a large number of candidate windows. However, if we apply only a limited number of template windows, the results would be unsatisfactory. This technique is known as sliding window technique to select Region of Interest (RoI). *Feature description*: Once the RoI is selected, we need to compute feature descriptors which provide robust and semantic representation of object. HOG [10], SIFT [11] and SURF [12] features are representative ones. However, due to the different appearance of various objects, illumination changes and occlusions, it is very difficult to manually design a feature extractor to perfectly represent all objects. *Classification*: Besides feature description, a classifier is required to distinguish an object class from other object classes. For instance, Support Vector Machine (SVM) [13] and Deformable Parts Model (DPM) [14] are some of the possible choices.

Papageorgiou *et al* [16] uses rectified Haar wavelets as feature descriptors along with SVM classifier for person detection. A direct approach of extracting edge images and matching them to a set of learned exemplars is given in [17]. Viola and Jones proposed boosted cascade of simple features to detect objects in [15]. The images are represented in a new representation called 'integral image', which is a fast way of calculating Haar-like features. Ada-boost algorithm is used in choosing features and improving performance. This technique is used to build a strong classifier with a cascade of many weak classifiers.

2.2 Deep learning methods

With the advent of deep learning, some of the most difficult problems in computer vision, those for which no formal models exist, started to have realistic solutions. Among those problems, object detection is also being treated with deep learning techniques [6]. Deep learning models are neural networks with deep architecture. Neural networks are inspired by the neural structure of the human brain and aim at solving learning problems in a systematic way. The emergence of the ImageNet database [18], increase in computational capabilities with GPUs and important advances in design of networks and training strategies has led to the development of deep learning in present millennium. Dropout and data augmentation have helped to decrease the problem of over fitting. Batch normalization (BN) [19] allowed to realize and effectively train very deep neural networks. Meanwhile, various network structures like AlexNet [6], VGG [20], ResNet [21], GoogleLeNet [22] have been studied to improve the performance of the feature extraction networks.

The framework for object detection can be classified into three types: 1) Sliding window detectors with neural networks [23]. 2) Traditional object detection pipeline, generating region

proposals at first and then classifying each region into object category. The region proposal methods include R-CNN [24], Fast R-CNN [25], Faster R-CNN [26], R-FCN [27], FPN [28], Mask R-CNN [29]. 3) Object detection as a classification problem with unified architecture to both classify and localize the objects. The classification based one stage methods include YOLO [30], YOLOv2 [31], YOLOv3 [32], SSD [33], DSSD [34].

The framework for object detection can be mainly classified into two types. One follows a two stage process of proposing regions and classification and other follows a single stage process as a unified architecture to both classify and localize.

2.2.1 Two-stage methods

Girshick *et al* [24] adopts a selective search [35] method to propose 2k RoIs in the image. A Convolutional Neural Network (CNN) is run on top of the proposed regions to extract more semantic and high-level feature representation of RoIs. A pre-trained SVM classifier is used to discern among the classes of objects and a linear bounding box regression is used to obtain tighter bounding boxes for localising the objects. Although selective search is computationally more economical than brute-force search by sliding window, it is still costly, as selective search takes 2 seconds to propose 2k RoIs. It is very costly to run CNN on top of each of 2k regions. The inference time is about 40 - 50 seconds. The R-CNN [24] is improved by Fast-RCNN [25] by incorporating mainly two augmentations. One is performing the feature extraction over image by CNN before proposing regions, thus only running one CNN over entire image instead of 2k CNNs. The other is to replace SVM with softmax layer, thus extending the neural network for prediction instead of creating new model. Except for region proposals, all parameters in this architecture are optimized via a multi task loss in an end-to-end way. One bottleneck still remaining in Fast R-CNN is selective search for RoIs generation, which is costly. The inference time for this approach is 2 seconds. Faster-RCNN [26] replaces the slow region proposal method of selective search by an internal deep network called Region Proposal Network (RPN), to improve the speed, thus the name Faster R-CNN. RPN is achieved with a fully convolutional neural network (which shares layers with object detection network) which has the ability to predict object boundaries and scores at each position simultaneously. The concept of anchor boxes or default bounding boxes is presented in [26]. The inference time for this approach is 0.2 seconds.

2.2.2 Single-stage methods

Two stage frameworks comprise of region proposal generation and classification (feature extraction with CNN, classification and bounding box regression), which are usually trained separately. Even in the end-to-end Faster R-CNN framework an alternative training is required to obtain shared parameters between RPN and detection network. As a consequence, the framework remains as a bottleneck for real time applications. Single stage frameworks based on global classification/regression, can directly map from image pixels to bounding box coordinates and class probabilities, thus reduces the time expense and work in real time applications. Redmon *et al* [30] proposed a network that looks at the image only once to detect multiple objects, thus the name YOLO. This model has a unified network to perform classification and localization of objects at once, thus end-to-end training of network can be achieved. Liu *et al* [33] formulated a network that predicts and localizes the objects in single shot. Multi scale feature maps with anchor boxes are used to obtain the predictions of objects of different sizes and aspect ratios.

3. AN OVERVIEW OF COMPARED METHODS

This section provides an overview of traditional (HOG and SVM) and deep learning methods (SSD and YOLO) for object detection.

3.1 Object detection with HOG and SVM

The general pipeline in traditional object detection is: RoIs selection, feature extraction and classification. There are many feature descriptors like SIFT [11], SURF [12], Haar-like [15], HOG [10] which can be used to extract features for classification. Dalal and Triggs [10] proposed HOG feature descriptor which has gained wider recognition as a successful feature descriptor for object detection. The HOG features provide robust, reliable and high-level semantic representation of image regions.

3.1.1 Region selection - Selective search

Given an image to detect the objects, firstly we need to select the image windows with potential possibility of object, also called as RoIs. Sliding window technique is one of the brute force approach to select image patches for object detection. Selective search [35] is a region proposal algorithm used in object detection. Selective search starts with considering individual pixels as their own group. Next, the similarity measure based on colour, texture, size and shape compatibility is calculated for each group and two closer ones are combined. The merging of the regions is continued until everything is combined. This method is designed to be faster than sliding window technique with a very high recall. The method of selective search is used for region proposal.

3.1.2 Feature extraction - modified HOG

HOG is a gradient based feature descriptor and it captures the information about the object's shape well. The local object shape within an image is characterized by the distribution of gradient magnitude and direction. In order to detect the objects in the image, we need to run and classify HOG descriptors for a large number of image patches. To accomplish this with less computation effort, we slightly modified the original computation of HOG [10]. The concept of integral image proposed by Viola and Jones [15] is used to leverage the computation speed for the calculation of HOG features. Firstly, we compute the gradients in horizontal and vertical directions of an image (of shape $m \times n$) and calculate the magnitude and direction of gradients of image. Secondly, we discretize the gradient orientations into q orientation bins and form a histogram of oriented gradients for each pixel, resulting in matrix of shape $m \times n \times q$. Finally, we compute the integral image of the computed HOG, so that we can compute the HOG for any image patch efficiently in constant time, this we call integral gradient image (IntGradImg). For instance, given a rectangular image patch represented by points (p11, p12, p21, p22) we can compute HOG descriptor as given in (1).

$$\text{hog} = \text{IntGradImg}(p11) - \text{IntGradImg}(p12) - \text{IntGradImg}(p21) + \text{IntGradImg}(p22) \quad (1)$$

For any given image patch, we subdivide the image patch into equal spatial cells (nx, ny). For each cell, we calculate the HOG descriptor which is discretized into q orientation bins as stated above. We finally concatenate the histograms of all the cells and normalize the resulting descriptor vector with respect to its L2-norm.

3.1.3 Classification - SVM

SVM trained on HOG descriptors is considered as a de facto standard in many visual perception tasks [36]. The idea of SVM is simple, to build an optimal hyperplane which separates the data into classes. SVM is built on the concept of margins, where margin is the separation between the closest class points called support vectors. The goal of the SVM classifier is to maximize the margins to result an optimal hyperplane to classify the data.

3.2 Object detection with SSD

SSD [33] detects the objects by passing through input image only once in a single shot, to predict object class for classification and bounding box for localization of objects. Concretely, for a given input image, firstly, the input image is passed through a series of convolutional layers yielding feature maps at different scales. Secondly, for each location of feature map a small (of shape 3×3) convolutional filter is used to evaluate a small set of default bounding boxes called anchor boxes. Finally, for each anchor box, the bounding box offsets and class probabilities are simultaneously predicted.

The architecture of SSD consists of feature map extraction network, additional feature layers and prediction layers. VGG16 [20] is used as feature map extraction network (also called backbone network) in [33]. In addition to *conv4_3* layer of VGG16, five additional feature layers as a part of multi-scale feature maps are used to detect objects. At each location of feature map, a certain number of anchor boxes are evaluated to predict the objects, and each prediction comprises of $(C+4)$ parameters, where C is the number of classes of objects and 4 are rectangular bounding box parameters. This type of prediction in multiple boxes is known as *multibox* detection. Instead of RPN as in Faster R-CNN [26], SSD uses 3×3 convolutional filters with $k \cdot (C+4)$ number of filters, where k is number of anchor boxes for each layer, to predict class and bounding box parameters. The predictions from multi scale feature layers are concatenated to obtain final predictions. Alongside VGG16, MobileNetV2 [8] and ResNeXt50 [9] are also used as backbone networks to evaluate SSD.

The objects of different sizes and aspect ratios are detected by SSD by leveraging the use of multi scale feature maps and default bounding boxes. Lower resolution feature maps with higher receptive field are responsible to detect bigger objects, on the other hand, higher resolution feature maps with lower receptive field are responsible for the detection of smaller objects. SSD uses non-max suppression to filter the duplicate predictions.

3.3 Object detection with YOLO

The main idea of YOLO [30] is that the input image is divided into $S \times S$ grid, if the centre of object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts k bounding boxes, confidence scores and conditional class probabilities. At inference time, in order to get class specific confidence scores for each box, we multiply conditional class probability and individual box confidence predictions. The predictions are encoded as an array of shape $S \times S \times k \cdot (5+C)$, where k is number of anchor boxes and C is classes. Classification and localization is done by single network, thereby end-to-end training is possible for better accuracy. YOLOv2 is proposed in [31] to make YOLO better, stronger and faster. YOLOv2 proposes certain improvements to YOLO with the objective to improve accuracy of detections, like employing batch normalization, high resolution classifier and anchor boxes. YOLOv3 [32] employs a feature pyramid like prediction network at 3 different scales to cope with objects of different sizes.

4. IMPLEMENTATION DETAILS

4.1 About the dataset

The traditional and deep learning object detection methods are compared by evaluating the methods on PASCAL VOC 2007 and 2012 dataset [7]. The dataset consists of 16,551 images in the PASCAL VOC 2007 and 2012 training and validation dataset and 4,925 images in the PASCAL VOC 2007 test dataset. The dataset consists of 20 classes of objects. For the evaluation of methods, we consider five classes namely, bicycle, bus, car, motorbike and person which are relevant for traffic scenarios and all other objects are treated as 'other' class.

4.2 Training details - HOG and SVM

In order to train a SVM classifier, we need to collect positive and negative samples for different classes. The positive training and validation samples of all classes are collected from the annotated training and validation images subset of PASCAL VOC 2007 and 2012 dataset respectively. The samples are described with HOG descriptor as specified in Section 3.1.2.

The SVM classifier is trained with training samples for six different classes (bicycle, bus, car, motorbike, person, other). The support vector classification with a linear kernel from Scikit-learn [37] is used, whose implementation is based on libsvm [38]. We use one-vs-rest multi class strategy to train the classifier. This strategy tries to fit one classifier per class. For each classifier, the class is fitted against all other classes. The main advantage of such strategy is its interpretability, where we can gain knowledge about the class by inspecting its corresponding classifier.

The classifier is validated on the validation samples and by experimentation the regularization parameter (C) is set to 0.92 to reduce over fitting of data. To obtain a better classifier, good negative samples are important. A technique known as bootstrapping is applied, to select hard negative samples for next round of training by evaluating the current classifier on negative samples and selecting high confident negative samples. The classifier is re-trained using original positive samples and original + new negative samples.

To detect objects in a given image, the image is passed through a detection module. The detection module consists of sub modules such as, selecting RoIs using selective search, computing integral gradient image as described in Section 3.1.2, describing each RoI using HOG descriptor, classification of the descriptor using the trained classifier. The detection module may predict multiple bounding boxes for a same object. The filtering of predictions is performed by applying per class confidence thresholding and non-maximum suppression. The method is implemented in Python with the aid of open-source Python libraries like Numpy [40] and Scikit-learn [37].

4.3 Training details - SSD

SSD is trained with 16,551 images with ground truth annotations from the PASCAL VOC 2007 and 2012 training and validation dataset on a NVIDIA GeForce RTX 2080 Ti GPU. The network is trained with VGG16 [20], MobileNetV2 [8] and ResNeXt50 [9] as backbones, that are originally trained on the ImageNet dataset [18]. In order to train the network, we need to choose the anchor boxes manually, depending on our dataset. Four or six anchor boxes are attached to each prediction layer of the network. We define scale value and aspect ratios of anchor boxes for each prediction layer of the network. The scale values range from 0.1 to 0.9 for higher resolution layers to lower resolution layers respectively. The aspect ratios for layers with six anchor boxes are: 1, 2, 3, 1/2 and 1/3. The width and height of anchor boxes are calculated using (2).

$$\begin{aligned} \text{width} &= \text{scale} \sqrt{\text{aspect ratio}} \\ \text{height} &= \text{scale} / \sqrt{\text{aspect ratio}} \end{aligned} \quad (2)$$

The assignment of anchor boxes per prediction layer results in large number of anchor boxes. The anchor boxes are classified as positive matches and negative matches. The SSD penalizes localization loss from only positive matched anchor boxes. An anchor box is considered positive match, if the Intersection over Union (IoU) of that corresponding anchor box with ground truth box is greater than 0.5. An anchor box whose IoU with ground truth box is greater than 0.3 but less than 0.5 are considered as neutral boxes and are not considered for computation of loss, as these boxes are 'too close' to a ground truth box to be a valid negative background box. The loss function is a weighted sum of localization and classification loss. The large number of anchor

boxes results in higher negative matches than positive matches, this results in class imbalance which hurts the training. So, hard negative mining is applied, where instead of using all negatives we sort the negatives by their confidence loss and pick the negatives with top loss, such that negative to positive matches ratio is kept at most as 3:1. This results in faster and stable training. Batch normalization and data augmentation schemes, such as flipping, cropping, colour distortions are applied to avoid over fitting. The SSD method is implemented in Python using TensorFlow library [41]. The network is trained for 120 epochs with 1000 steps per epoch, with a batch size of 32. Adam optimizer is used with a learning rate of $1e^{-3}$ for first 80 epochs and $1e^{-4}$ for rest of epochs.

4.4 Training details - YOLO

YOLO is trained with 16,551 images with ground truth annotations from the PASCAL VOC 2007 and 2012 training and validation dataset on a NVIDIA GeForce RTX 2080 Ti GPU. The network is trained with MobileNetV1 [39] and ResNet50 [12] as backbones, that are originally trained on the ImageNet dataset [18]. Three anchor boxes are attached to each of the three feature map layers of the network. k-means clustering technique is used to find the anchor boxes. Batch normalization and data augmentation schemes, such as flipping, cropping, colour distortions are applied to avoid over fitting. The YOLO method is implemented in Python using TensorFlow library [41]. The network is trained for 100 epochs with 1000 steps per epoch, with a batch size of 16. Adam optimizer is used with a learning rate of $1e^{-3}$ for initial 30 epochs and $1e^{-5}$ for rest of epochs.

5. RESULTS AND DISCUSSION

This section provides a comparison of qualitative and quantitative results of the object detection task by traditional and deep learning approaches and discusses about the critical aspects of the two approaches.

5.1 Qualitative Results

The results of object detection using traditional and deep learning approaches is shown in Figure 2. As shown in Figure 2b, SSD is able to detect small objects (persons and cars) in the background, but in Figure 2e YOLO is unable to detect the small objects. SSD predicts the objects using multi scale feature maps (6 scales), so the receptive fields of multi layer feature maps are able to detect objects of small and big sizes. The SSD with VGG16 network predicts the persons in the background as shown in Figure 2b(top) but SSD with MobileNetV2 network is unable to detect the persons at the background as shown in Figure 2a(top). MobileNetV2 [8] uses depthwise separable convolutions, that consists of a depthwise and pointwise convolutions one after another as opposed to normal convolutions in VGG16. This drastically reduces the number of parameters of the network, thereby reducing model size and complexity, resulting in increase of processing speed but at the cost of small reduction in accuracy as shown in Figure 2a (top) and Figure 2b(top).

The traditional object detection approach (HOG+SVM) could not detect objects with good accuracy as compared to deep learning approaches. As shown in Figure 2f, the selective search which is used to generate regions is not good enough to provide more regressed bounding boxes like in deep learning approaches. The traditional approach which uses SVM as a classifier is unable to provide better classification results when compared to deep learning approaches.

5.2 Quantitative Evaluation

The evaluation of traditional and deep learning object detection approaches is performed on PASCAL VOC 2007 test set. The mAP is considered as an evaluation metric for object detection. mAP is the average of precision over all categories. Average Precision (AP) is the

area under Precision-Recall curve. A prediction is considered positive, if its IoU with the ground truth is greater than 0.5.



Figure 2. Object detection results: (a) SSD-MobileNetV2 (b) SSD-VGG16 (c) SSD-ResNeXt50 (d) YOLO-MobileNetV1 (e) YOLO-ResNet50 (f) SVM-HOG

The mAP, per-class average precision and speed (in frames per second - fps) of traditional and deep learning approaches is shown in Table 1.

The deep learning approaches like SSD and YOLO outperform the traditional approach like HOG and SVM by huge margin in terms of mAP as shown in Table 1. The inference of deep learning approaches is performed on a NVIDIA GeForce RTX 2080 Ti GPU. The processing speeds of both the deep learning approaches are above par real time speeds, which is very crucial for self-driving vehicles in traffic scenarios. Although SSD with ResNeXt50 network gives higher mAP than other SSD backbone architectures, the processing speed for this approach reduces to almost half as compared with SSD with MobileNetV2 architecture. As a trade off between accuracy and speed, we can consider SSD with VGG16 as a better approach for object detection.

Table 1. Mean Average Precision (mAP), per-class Average precision (AP) and processing speed of different object detection approaches on PASCAL VOC 2007 test set

Method	Backbone	Parameters	Per-class AP					mAP	Speed (in fps)
			Bicycle	Bus	Car	Motorbike	Person		
SSD	VGG16	~26.3M	84.7	84.5	81.6	81.9	75.9	81.7	71
	MobileNetV2	~7.5M	81.4	81.1	78.1	78.6	72.7	78.4	95
	ResNeXt50	~29.5M	86.2	84.2	82.2	81.7	77.3	82.3	48
YOLOv3	MobileNetV1	~24.2M	75.9	75.7	73.5	73.8	68.1	73.4	73
	ResNet50	~45.3M	79.8	79.4	76.4	77.1	71.3	76.8	33
SVM	HOG	-	24.5	23.7	21.2	20.7	17.8	21.6	2*

*on Intel Core i9 CPU

5.3 Discussion

Deep learning is pushing its limits to obtain a super human accuracy in object detection when compared to traditional object detection methods. There are benefits and drawbacks to deep learning approaches as compared to traditional computer vision techniques. Deep learning brings in many challenges such as computation power, big data, training time etc. But rapid progressions in device capability in terms of computation power, memory capacity has improved the performance and cost effectiveness of deep learning approaches. The rise of data, where in the era itself is being called as 'era of data', has also paved way for the up rise of deep learning approaches.

Traditional methods of computer vision requires expert analysis, for instance, choosing a feature descriptor for classification or correspondence matching. It is difficult to choose which features are significant for a specific data. For example, in case of classification, as the number of classes increases the feature extraction will become more cumbersome. The handcrafted feature design remains a bottleneck in traditional methods, as it is very subjective in nature. On the other hand, deep learning algorithms are trained end-to-end to extract features ranging from low to high level from the data. The deep learning network discovers underlying semantic information and automatically learns the most salient features, which provides a better accuracy compared to handcrafted feature descriptors.

Albeit as shown in Table 1, it is established that deep learning methods perform well than traditional methods for object detection, there are trade-offs with respect to computing requirements, availability of data and training time. The training of deep neural networks require lots of computation power and training time. In our case, to train a SSD model, it took nearly 24 hours on a NVIDIA GeForce RTX 2080 Ti GPU. The deep learning methods require lots of data to train the network otherwise the network may overfit to training data and may not generalize well to other data. The features learned from a neural network is specific to the trained dataset, if not trained well, probably won't perform well for other images. Whereas, some traditional algorithms like SURF can be used for applications such as correspondence matching which don't require class specific knowledge. So, deep learning is not a unique solution for every problem but in our case of object detection, it outperforms the traditional techniques.

6. CONCLUSION

This work presents a comparison of traditional and deep learning approaches for the task of object detection in traffic scenarios, in terms of mAP and processing speed. The traditional

method of handcrafted feature descriptor like HOG and a linear SVM classifier is compared with deep learning algorithms like SSD and YOLO. SSD algorithm is implemented with different backbone architectures like VGG16, MobileNetV2 and ResNeXt50, similarly YOLO algorithm is implemented with MobileNetV1 and ResNet50 architectures to compare the performance of the approaches. The training of the algorithms is performed on PASCAL VOC 2007 and 2012 training and validation datasets and inference is performed on PASCAL VOC 2007 test set. For the calculation of mAP, we consider five classes namely, bicycle, bus, car, motorbike and person, which are relevant for traffic scenarios. The training and inference is performed on a NVIDIA GeForce RTX 2080 Ti GPU.

We presented both qualitative and quantitative results of the object detection for comparison of traditional and deep learning approaches in Section 5. Qualitatively and quantitatively, the deep learning approaches outperform the traditional approach both in accuracy and speed, as shown in Figure 2 and Table 1 respectively. The deep learning approaches obtain above par real time processing speed, which is significant for self-driving vehicles in traffic scenarios. SSD algorithm is able to detect the small objects in the background (persons and cars in Figure 2b), which is not the case with YOLO, this is due to the fact that SSD uses multi scale feature maps which leverages small and large receptive fields to detect small and large objects. SSD with MobileNetV2 architecture obtains higher processing speed due to depthwise separable convolutions which drastically reduces the number of parameters in the network, but this reduces the mAP compared to ResNeXt50 architecture. As a trade off between accuracy and speed, one can consider SSD with VGG16 architecture to perform well.

In the traditional approaches, we use handcrafted features like SIFT, HOG etc. for applications related to visual perception. The selection of handcrafted features is very subjective in nature and it becomes a cumbersome task to select the feature descriptors when a large variety of subjects are involved in the scene. On the other hand, deep learning architectures learn the underlying patterns in the data and automatically work most relevant and salient features from the data, which leverages the accuracy in certain computer vision applications (like object detection). But, this is achieved at the cost of requiring lot of data and time to train the network alongside with high computational power.

ACKNOWLEDGEMENTS

This project is funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 765866.

REFERENCES

- [1] Pettitt Michael, Burnett Gary and Stevens Alan., "Defining Driver Distraction," in *Intelligent Transportation Society of America - 12th World Congress on Intelligent Transport Systems*, 2005.
- [2] Singh J, Sahni MK, Bilquees S, Khan SMS, Haq I, "Reasons for road traffic accidents - victims' perspective," in *Int. J. Med. Sci. Public Health*, 2016;5:814-818.
- [3] Zhang Xingjian, Zhao Xiaohua, Du Hongji and Rong Jian, "A Study on the Effects of Fatigue Driving and Drunk Driving on Drivers' Physical Characteristics. Traffic injury prevention," 15.10.1080/15389588.2014.881996, 2004.
- [4] W. Jones, "Keeping Cars from Crashing," in *IEEE Spectrum*, vol. 38, no 9, pp. 40-45, 2001.
- [5] L. Li, D. Wen, N.-N. Zheng and L.-C. Shen, "Cognitive cars: A new frontier for ADAS research," in *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 395-407, Mar. 2012.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

- [7] Everingham M, Van Gool L, Williams C K I, Winn J. and Zisserman A., "The PASCAL Visual Object Classes (VOC) Challenge," in *International Journal of Computer Vision*, 88(2), 303-338, 2010.
- [8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 4510-4520.
- [9] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 5987-5995.
- [10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
- [11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] Herbert Bay, Tinne Tuytelaars and Luc Van Gool, "SURF : Speeded Up Robust Features," in *ECCV*, 2006.
- [13] C. Cortes and V. Vapnik, "Support vector machine," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 1627–1645, 2010.
- [15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, pp. I-I. doi: 10.1109/CVPR.2001.990517.
- [16] C. Papageorgiou and T. Poggio, "A trainable system for object detection," in *IJCV*, 38(1):15–33, 2000.
- [17] D. M. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *CVPR*, Fort Collins, Colorado, USA, pages 87–93, 1999.
- [18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
- [19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.
- [20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.
- [23] H. Nakahara, H. Yonekawa, S. Sato, "An object detector based on multiscale sliding window search using a fully pipelined binarized CNN on an FPGA," in *Proc. IEEE Int. Conf. Field Program. Technol. (ICFPT)*, pp. 168-175, Dec. 2017.
- [24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
- [25] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.
- [26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.
- [27] Y. Li, K. He, J. Sun et al., "R-fcn: Object detection via region-based fully convolutional networks," in *NIPS*, 2016, pp. 379–387.

- [28] T.-Y. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.
- [29] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, "Mask r-cnn," in *ICCV*, 2017.
- [30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.
- [31] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv:1612.08242*, 2016.
- [32] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767*, 2018.
- [33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.
- [34] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv:1701.06659*, 2017.
- [35] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," in *Int. J. of Comput. Vision*, vol. 104, no. 2, pp. 154–171, 2013.
- [36] H. Bristow and S. Lucey, "Why do linear SVMs trained on HOG features perform so well?" in *arXiv preprint arXiv:1406.2419*, 2014.
- [37] Pedregosa et al., "Scikit-learn: Machine Learning in Python," in *JMLR 12*, pp. 2825–2830, 2011.
- [38] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM : a library for support vector machines," in *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [39] Andrew G. Howard, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861v1*.
- [40] 1. Oliphant TE, "A guide to NumPy," Vol. 1. *Trelgol Publishing USA*, 2006.
- [41] Martin abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015.

Authors

Gopi Krishna Erabati is a Marie-Curie early stage researcher and doctoral student at Institute of Systems and Robotics (ISR), University of Coimbra, Portugal. He received his bachelor's degree in 2013 from Kakatiya University, India and MSc in 2018 from University of Dijon, France. His research interests include computer vision, machine learning, visual perception and autonomous navigation.

Nuno Gonçalves is a researcher at the ISR, University of Coimbra and Tenured Assistant Professor at the Dept. of Electrical and Computers Engineering of the University of Coimbra. He received his MSc and PhD degrees in 2002 and 2008, respectively from the University of Coimbra. His main research areas are computer vision, computer graphics and machine learning, with special emphasis to geometric problem in vision systems.

Hélder Araújo is a Full Professor at the Dept. of Electrical and Computer Engineering of the University of Coimbra and senior researcher at the ISR, University of Coimbra. His research interests include robot vision, robot navigation, computer vision, sensor fusion and cognitive robotics.

USING HOLOGRAPHICALLY COMPRESSED EMBEDDINGS IN QUESTION ANSWERING

Salvador E. Barbosa

Department of Computer Science, Middle Tennessee State University,
Murfreesboro, TN, USA

ABSTRACT

Word vector representations are central to deep learning natural language processing models. Many forms of these vectors, known as embeddings, exist, including word2vec and GloVe. Embeddings are trained on large corpora and learn the word's usage in context, capturing the semantic relationship between words. However, the semantics from such training are at the level of distinct words (known as word types), and can be ambiguous when, for example, a word type can be either a noun or a verb. In question answering, parts-of-speech and named entity types are important, but encoding these attributes in neural models expands the size of the input. This research employs holographic compression of pre-trained embeddings, to represent a token, its part-of-speech, and named entity type, in the same dimension as representing only the token. The implementation, in a modified question answering recurrent deep learning network, shows that semantic relationships are preserved, and yields strong performance.

KEYWORDS

Question Answering, Vector Embeddings, Holographic Reduced Representations, DrQA, SQuAD

1. INTRODUCTION

Question answering (QA) is the natural language processing (NLP) task of selecting a span of text from one or more context documents, as the answer to a specific question. The context documents, referred to as *passages* in this paper, may be anything from a single sentence to a collection of multiple documents. Early QA systems were symbolic and rule-based, but recently, deep neural networks have become the standard for such systems, yielding excellent performance while avoiding the brittleness of rule-based systems. The distributed representations learned in these networks have been significantly enhanced by vector representations of word tokens, known as embeddings.

Embeddings are dense, continuous vectors, learned from text corpora, which capture the semantic relatedness of individual words from the context of surrounding words. It should be noted that the relationship extracted here is not necessarily a synonymic one, but is rather one of being in the company of similar words. Thus, the words *love* and *hate* are usually identified as highly related embeddings, even though they are antonyms of each other, because they tend to appear in similar contexts.

Commonly used embeddings in NLP include *word2vec* [1] and *GloVe* [2]. These embeddings are pre-trained on large corpora (such as the Wikipedia text), and represent tokens as real-valued vectors with 50 to 300 dimensions. Deep neural networks utilize vectors of individual tokens to

represent words, phrases, sentences, and higher aggregations of text, depending on the application and underlying network topology. Embeddings are mostly computed at the level of distinct words (known as word types). As such, there is a single vector for a word, regardless of whether it may have more than one part-of-speech in its various uses in the training corpus. Because of their importance to the task, many QA systems use the part-of-speech tag and named entity type (Person, Organization, Money, Time, etc.) of individual words as separate features, in extracting an answer. Doing so, however, expands the dimension of the input for each token, potentially leading to the curse of dimensionality. This research eliminates the expansion in the size of the input by using holographic compression on the vectors. Employing the proposed approach, the part-of-speech tag and named entity type of each token is encoded without increasing the size of the input, while providing the benefits generally associated with dimensionality reduction: a decrease in overfitting and a reduction in processing time.

Holographic reduced representations (HRR) were devised by Tony Plate as a means for combining two or more vectors of a given dimension, into a single vector having the same dimension as only one of those components [3]. Thus, unlike the concatenation of two vectors of dimension n which results in a vector of dimension $2n$, HRR uses circular convolution to combine vectors, without increasing the dimensionality of the resultant vector.

Recurrent neural networks (RNN) are commonly used to learn from sequences of data, such as the words that form a sentence or larger collection of words [4]. Rather than feeding the entire sequence as a single input (concatenated or superimposed in some way), individual elements of the sequence are processed one at a time. The RNN is constructed so that at each time step, it considers the current input element, along with a combined or accumulated representation of the elements that preceded it, to update its learning. The long-short-term-memory (LSTM) architecture has become ubiquitous in question answering applications in recent times [5] – [7].

DrQA is a question answering framework developed by Facebook Research, to answer open domain questions from sources such as Wikipedia [8]. It is made up of two major components: A piece to retrieve relevant documents for the question posed, known as the Document Retriever, and the Document Reader, a multi-layer recurrent neural network that extracts the answer span from the retrieved documents. The Document Reader's input are individual tokens (represented as 300-dimensional vectors), along with their part-of-speech tag and named entity type (encoded as inputs of 50 and 19 elements respectively).

This paper describes experiments carried out to evaluate the use of HRR to compress each token, its part-of-speech, and its named entity type (if any), into a single 300-dimensional vector. Compared to DrQA, this decreases the width of the input data from 369 to 300, a reduction of over 18%, thereby speeding up training. In Section 2, the background and prior related work is described. Section 3 presents the holographic reduction process, including the slot and filler scheme employed, and describes the modifications made to the Document Reader portion of DrQA, to enable it to process the holographically compressed tokens. The framing of the experiments are found in Section 4, with results and analysis are given in Section 5. The conclusions and future research avenues are summarized in Section 6.

2. BACKGROUND AND RELATED WORK

This research is based primarily on two prior works: holographic reduced representations, and the DrQA question answering architecture. Each of these is described in turn below.

2.1. Holographic Reduced Representations

Holographic reduced representations (HRR) apply algorithms from the field of signal processing, namely circular convolution and its approximate inverse, circular correlation, to compress multiple vectors of dimension n , into a single vector having that dimension [3]. In order to accomplish this, elements of the initial vectors are superposed in the resulting vector. While superposition can also be achieved by averaging two vectors element-wise, the average operation is not invertible. Circular convolution, on the other hand, also results in a superposed vector, but is an invertible operation. Circular convolution applied to vectors \mathbf{c} (having elements c_0, \dots, c_n) and \mathbf{x} (with elements x_0, \dots, x_n), yields a vector \mathbf{t} , having elements t_0, \dots, t_n . Figure 1 shows the circular convolution compression of two vectors (c and x), each having three dimensions, into a single three-dimensional vector, t , whose elements are computed as depicted.

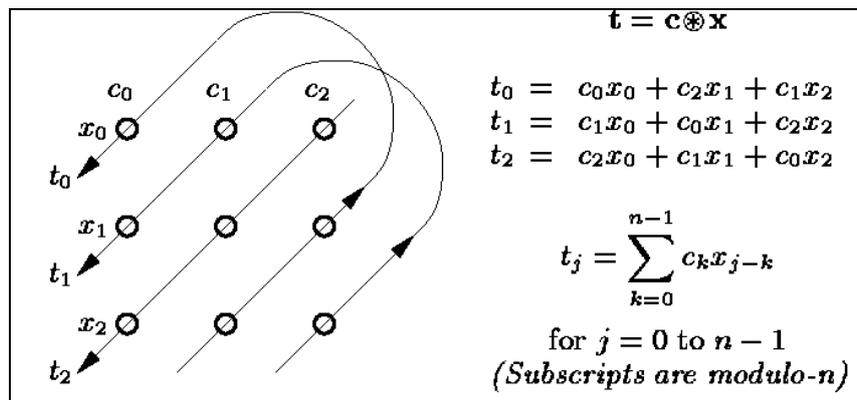


Figure 1. Circular convolution (adapted from [3]).

Multiple holographically compressed vectors may be composed together by element-wise summing or averaging. Thus, the composition operation enables compression of more than two vectors into a single vector having the same dimension as a single constituent component.

Circular convolution is a commutative operation, and as such cannot be used to represent complex structures, such as sequences or hierarchies, since order is lost. A recommended approach for representing a complex structure such as a frame with slots, is to create a frame label vector, and sum it with one or more slot label vectors, circularly convolved with their respective filler vectors [3]. Various aspects of working with HRR vectors require that they be (nearly) orthogonal. Pre-trained word vector embeddings do not necessarily have this orthogonality property.

This research effort makes use of a holographically reduced frame and slot structure, to represent the token, its part-of-speech tag, and named entity type. The approach followed results in quasi-orthogonal vectors, and is fully described in Section 3.

2.2. DrQA

The DrQA question answering framework developed by Facebook Research [8] was designed to answer open domain questions from large sources of information like Wikipedia. It is divided into two primary components: the Document Retriever retrieves documents (from Wikipedia or other sources) that may contain the answer to the question; the Document Reader identifies the answer span from the retrieved text documents. In the original work, this latter component was evaluated using SQuAD 1.1, a dataset consisting of a number of passages and over 100,000

questions, whose answers are extractable from the passages as spans of text [9]. The experiment described in this paper only makes use of the Document Reader and the SQuAD dataset. The Document Retriever is not considered.

The neural network used in the Document Reader is a composite of multiple neural networks, each having a distinct function. Tokens in both the question and the passage are encoded and presented to the network. The **question** encoding simply takes the 300-dimensional *GloVe* embedding for each token, passes it through a bidirectional LSTM, and combines the vector from the hidden units via a weighted sum (whose weights are learned by the network). The **passage** encoding is more complex, with each token represented by:

- Its 300-dimension *GloVe* vector
- Its part-of-speech tag
- Its named entity type (if any)
- A 4-element feature set, indicating: 1) whether the token matches a token in the question exactly; 2) whether there is a case-insensitive match between the token and a token in the question; 3) whether there is lemma form match between the token and a token in the question; and 4) the token's term frequency within the passage.

The above described representation for each passage token, and an attention-based real-valued weighted measure (derived from a simple feed-forward network) which provides the similarity between related but morphologically dissimilar words in the passage and question, such as *hit* and *single* in the music (or baseball) sense, is run through a bidirectional LSTM. That output, along with the weighted sum question representation, is fed to two bilinear attention classifiers: one predicts the probability of the passage token being the start of the answer span, and the other predicts the probability of it being the end of the answer span. The network's output is a pair: the passage token positions having the highest probabilities for being the start and end of the answer span. The architecture of the Document Reader neural network is shown in Figure 2, and depicts the partial representation of the question *When did Beyoncé release Dangerously in Love?* and the portion of the passage containing the answer, that reads in part ... *Beyoncé's debut album, Dangerously in Love, ...*

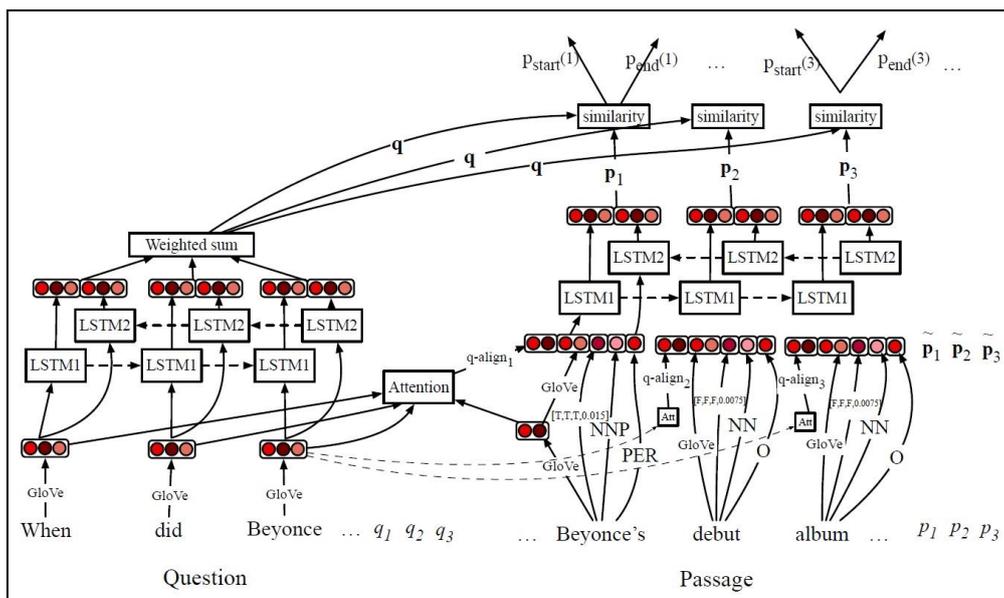


Figure 2. DrQA Architecture (adapted from [10]).

2.3. Related Work

A search of literature yielded no instances of works that use HRR with DrQA, thus establishing the novel aspect of this research. Only one prior work in general question answering utilizing HRRs with neural networks was found [11]. In those experiments, the goal was to detect the relationship between questions and answers (no mention of passages or context documents is made). Their architecture differs significantly from this research, first passing pre-trained embeddings for questions and answers into separate LSTMs, and passing the last hidden state from each to a layer where a holographic representation is constructed for further processing. An additional dissimilarity from this work is that circular correlation was used in constructing the holographic representation, rather than circular convolution.

The Document Reader architecture was modified for this research as shown in Figure 3. The passage input was changed to remove the previously separate encodings of parts-of-speech and named entity types. These attributes are now compressed into the token embeddings. The question input stage was altered to accept the compressed tokens which include part-of-speech and named entity type, unlike in the original architecture. All other aspects were retained.

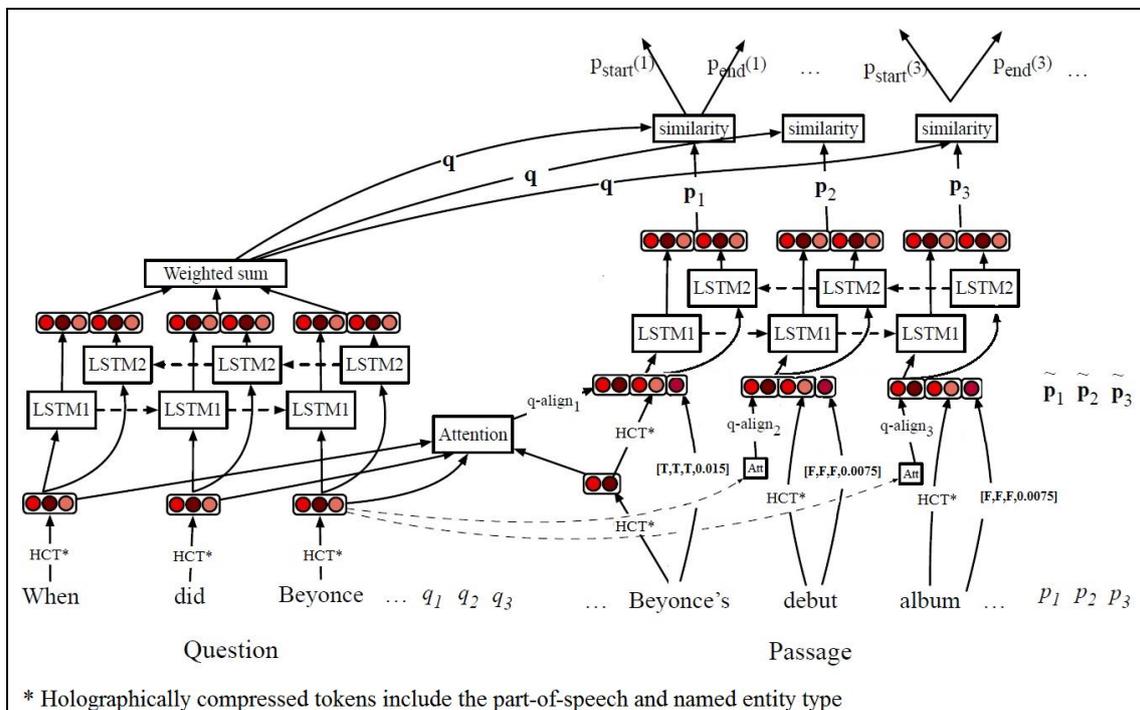


Figure 3. Modified Architecture.

3. METHODOLOGY

The experiments crafted for this research were designed to answer specific research questions:

Research question #1 – Can pre-trained word vector embeddings (which are not guaranteed to be orthogonal to each other) be made (nearly) orthogonal by constructing holographic frame and slot structures with them? The answer to this question is important, as the theory of holographic reduced representations relies on vectors being orthogonal to each other.

Research question #2 – Do holographically compressed token embeddings retain the semantic relationships they possessed pre-compression? The value of embeddings pre-trained on large corpora comes from the semantics that are learned from the various contexts in which words appears. To take advantage of this benefit, those relationships must generally be preserved.

Research question #3 – How do compressed tokens (consisting of a holographic frame-slots structure for the token, its part-of-speech, and its named entity type) perform, compared to uncompressed embeddings with separate features for part-of-speech and name entity type, in the DrQA Document Reader architecture? To hold promise for future applications, the accuracy metrics of the proposed compression approach must be close or superior to those of the unmodified method.

The methodology required work along two fronts. First, the pre-trained *GloVe* embeddings had to be converted into their new holographically compressed embeddings form. After this was done, the Document Reader architecture was adapted to accept the new input format. The approach followed in each of these phases is detailed in the subsequent two sections.

3.1. Holographically Compressed Tokens

The method undertaken to compress a word token was to create a frame and slots structure, as recommended in the seminal work [3]. In order to do this, several 300-dimensional vectors, were randomly generated to represent labels for the frame and each slot, as well as each part-of-speech tag and each named entity type. The structure was then constructed from the pre-trained *GloVe* 300-dimensional token vectors, and vectors for the frame label (HRR TOK), the token slot (TOK), the token’s part-of-speech tag slot (POS), and the token’s named-entity type slot (ENT), if one exists. The holographically compressed token (TOK_{HC}) in this work is calculated as:

$$TOK_{HC} = (HRR TOK + TOK \otimes token + POS \otimes pos_tag + ENT \otimes ne_type) / m$$

where *token*, *pos_tag*, and *ne_type* are vectors for the fillers: the token, its part-of-speech, and named entity type (if applicable), and \otimes is the circular convolution operator. The value *m* is the number of vectors summed (in the above example four vectors are summed). The division by this factor normalizes the compressed vectors. Tokens that do not have a named entity type are normalized by dividing by three). As an example, the token *IBM* would be represented as:

$$IBM_{HC} = (HRR TOK + TOK \otimes \mathbf{IBM} + POS \otimes \mathbf{NNP} + ENT \otimes \mathbf{ORG}) / 4$$

The italicized and bolded fillers above are the vectors for the token (*IBM*), its part-of-speech (*NNP* is the tag for proper noun), and named-entity type (*ORG* = organization). The vectors for the frame and slot labels, each part-of-speech tag, and each named entity type were randomly selected from a normal distribution with mean 0 and variance $1/n$, where *n* is the dimensionality of the vectors (300 in this case). While reduced holographic representations are generally presented with vectors having dimensionality in the thousands, in this study the randomly drawn vectors with 300 dimensions were sufficiently orthogonal, and their composition with the pre-trained *GloVe* embeddings yielded a vocabulary represented by vectors nearly orthogonal to each other, with the angle separating most pairs of vectors being greater than 75 degrees.

3.2. DrQA Document Reader Modifications

The source code for DrQA is available at <https://github.com/facebookresearch/DrQA>, under a BSD license. Since only the Document Reader was to be used, a version modified to run that

component stand-alone was downloaded from <https://github.com/hitvoice/DrQA>. This baseline uses the spaCy natural language processing framework (<https://spacy.io/>) to determine parts-of-speech and named entity types (the original uses the Stanford CoreNLP suite). *GloVe* embeddings of 300 dimensions, trained on the 840 billion word corpus, were used.

The source code was downloaded and modified to accommodate the holographically compressed token format. Changes were made to:

- The preprocessing stage (this is where grammatical annotations are made and the vocabulary of the SQuAD dataset is mapped to embeddings) – Since the compressed tokens include the part-of-speech tag and named entity type, this meant that a simple word type might no longer be distinct, as those attributes might be dissimilar in different contexts. The raw token was constructed from the word type (in lowercase), concatenated with its part-of-speech and named-entity type. As an example, the word type *fish* would be converted in the sentences below to the word type shown in parentheses:

I like to fish early in the morning. (fishVB) - *fish* is a verb

That was the best fried fish I have ever eaten. (fishNN) – *fish* is a noun

This is Dr. Fish. (fishNNPPERSON) – *fish* is a proper noun with type PERSON

While *fish* has a single pre-trained *GloVe* embedding, it maps to distinct compressed vectors in the above examples. Additionally, no tokens are left without representation in this research, as tokens that were not in *GloVe* were mapped to a special vector for unknown word types. These two facts lead to an increase in the size of the vocabulary of the holographically compressed tokens. In the experiment, the size of the vocabulary grew from 91,590 to 158,454 distinct tokens. Each modified token was used as an index that maps it to its holographically compressed vector, which is fed to the neural network.

The other change made to the preprocessing stage was the addition of holographic compression for question tokens. It should be noted that this differs from the baseline model, which does not use part-of-speech and named entity types for question tokens. However, this change was made to keep the representations coherent for the attention mechanism in the network.

The 4-tuple feature set of binary similarities and term frequency was retained for processing without modification.

- The neural network – Changes were made to the neural network to remove the paths for input of the explicit part-of-speech tag and named-entity type features associated with passage tokens, and to set it up to process only the compressed tokens and the 4-tuple. No changes were required to the paths that process the question tokens, as those simply accept a 300-dimensional vector. The difference in this case is that the holographically compressed vectors for the question tokens also contained the part-of-speech tag and named-entity type.

4. EXPERIMENTS

The experiments were constructed to facilitate a valid comparison between the DrQA Document Reader in its original form (using only the pre-trained *GloVe* embeddings) and the holographically compressed tokens constructed in this research. This included following the model validation approach undertaken for DrQA.

4.1. SQuAD Input Data

The SQuAD 1.1 question answering dataset [9] was used for this research. In this dataset, each answer is a span from the passage (or context) from which the question is taken. The training set used was made up of over 87K questions, and the validation data had over 10K questions.

4.2. Experimental Setup

The experiments were carried out in the same manner as was done for DrQA, including the validation approach. The experiments were run twice: once for the baseline (unmodified) Document Reader, and once using the holographically compressed tokens. Each experiment ran for 40 epochs.

4.3. Model evaluation and data collection

An evaluation of the model was completed using the validation set, at the end of each epoch, as was done in the original work. For consistency, two accuracy measures were collected at the completion of all epochs: an exact match metric (the predicted span was correctly identified with both the start and end indices), and a broader F1 measure for when the answer was correct, but incomplete (e.g., the correct answer spanned the string *John Smith*, but the model selected the span for either *John* or *Smith*).

5. RESULTS AND ANALYSIS

This section contains the results and analysis of the experiments. The findings are presented as they relate to each of the three research questions, in sections 5.1 through 5.3 respectively.

5.1. Orthogonality of the Holographic Vocabulary Vectors

In order to compose complex structures, such as sequences that represent a sentence, the resulting holographically compressed tokens, must be (nearly) orthogonal. While the experiments done here do not represent phrases or sentences, this effort was undertaken as a step toward that capability. Given the size of the final vocabulary, a full pairwise set of distances between the vectors was too computationally expensive to assess. Instead, to evaluate the orthogonality of the embeddings, two disjoint lists of 100,000 tokens were selected at random, and the cosine similarity for tokens at the same index position in those lists was calculated. In nearly 94% of these cases, the cosine similarity was less than 0.25 (less than 15 degrees from fully orthogonal). Since orthogonality is not an attribute of the *GloVe* embeddings, this characteristic is directly attributable to the holographic compression attained through the orthogonality of the frame and slot labels vectors used.

5.2. Preservation of Semantic Relationships

As was previously mentioned, the benefit of embeddings trained on large corpora is the robust set of semantic relationships extracted from word usage in context. It would be desirable to preserve these semantic relationships to the greatest extent possible. In the experiments, vectors were mapped to different vector bases by the holographic reduction process, making it impossible to retain the exact relationships that existed between tokens prior to their transformation. The approach taken in evaluating the preservation of semantic relationships was to select core tokens randomly, and to analyze the composition of their neighborhoods (which tokens are closest it?) in both bases. In each of the next figures, the core token is encircled and

its ten closest neighbors in both spaces are shown. The data is visualized in two dimensions by use of the t-SNE technique [12].

Figure 4, depicting the token *radially*, shows a case where the semantic positions are largely preserved, and only one token differs between the methods. The holographically reduced neighborhood does not include the embedding for *actuating*, adding in its stead a different, but semantically related, token (the embedding for *protrusion*).

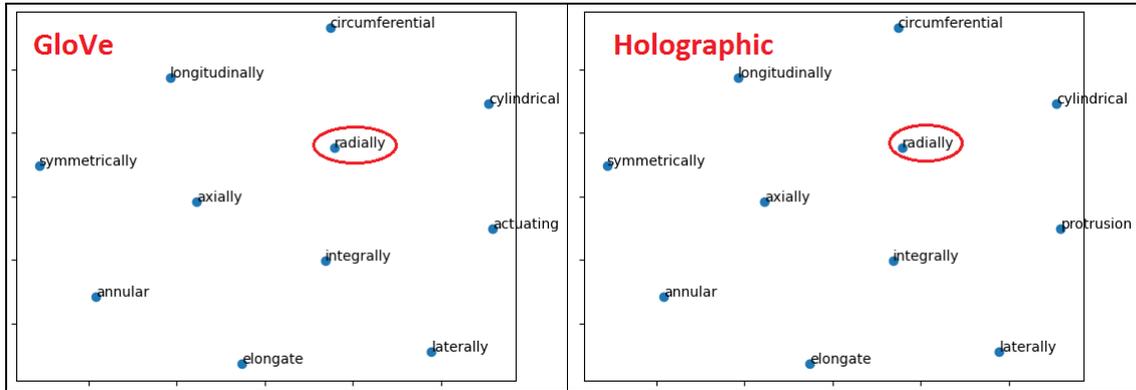


Figure 4. Vector neighborhoods for the token *radially*.

The neighborhood for the token *sprint* is shown in Figure 5. Most neighbors are similar in the two cases, with tokens *Sprint* and *sprints* holding the same relative positions. However, the relationships of many neighbors shifted (*javelin*, *moto*, *race*, *racing*, *sprinter*, and *sprinting*). The holographic embeddings omit tokens *triathlon* and *racer*, and include *champion* and *relay*.

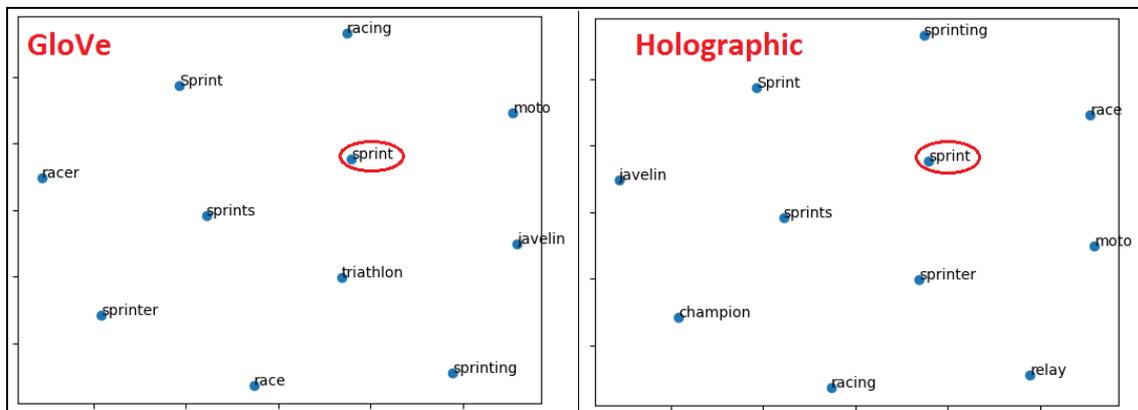


Figure 5. Vector neighborhoods for the token *sprint*.

Finally, Figure 6 portrays the neighborhoods for token *shared*. This was a case where there were significant differences between the two sets of embeddings. Only four neighbors are in common, with two holding their relative positions (*share* and *sharing*), and two showing some displacement (*own* and *Shared*). Tokens *both*, *communal*, *experiences*, *family*, *personal*, and *together* do not appear in the holographic neighborhood, which instead includes embeddings for *access*, *contributed*, *ideas*, *linked*, *others*, and *realized*.

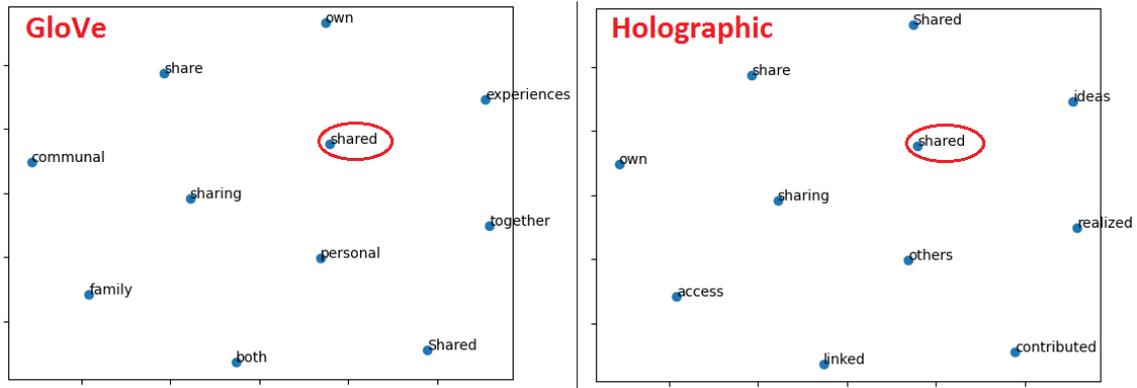


Figure 6. Vector neighborhoods for the token *shared*.

A detailed pairwise analysis was done on the composition of the neighborhoods. Specifically, the ten neighbors were sorted by their distance from the core token in both the unmodified and holographically compressed cases. The results showed that approximately: 18% of the vectors in each neighborhood were in the same relative position in both cases; 43% were present in both neighborhoods, but were shifted in distance; and 39% of words in the neighborhoods were disjoint between the two sets. In the disjoint cases, the departures seen were deemed reasonable by a manual review. Although the words that differ are semantically related to the core token, additional research is needed to better characterize these differences and their potential impact on applications.

5.3. Accuracy Metrics

Training took place over 40 epochs for each case, and the best model was selected based on the F1 score. For the unmodified Document Reader, F1 was 78.23%. That model yielded an exact match score of 68.83%. These values are consistent with those reported in the original DrQA research [8]. The holographically compressed approach's best model had an F1 measure of 76.28%, with exact match at 66.70%. The scores are summarized in Table 1.

Table 1. Summary of Accuracy Metrics.

	F1	Exact Match
Unmodified Document Reader	78.23%	68.83%
Holographically Compressed Tokens	76.28%	66.70%

While the performance attained fell short of the original work, it did not do so by much: both the exact match measure and the F1 metric were around two percentage points below the level for DrQA Document Reader. Furthermore, this research effort mirrored the approximately 10% difference seen between these two measures in the original case. As an initial effort without detailed optimization, the results point to a promising method that has the potential to be refined for better results.

On the positive side, the time for completion of individual epochs decreased by over 30% in the holographically compressed tokens case, due to the reduction in the input data width. Notwithstanding the added 22 minutes of preprocessing time (a one-time cost) incurred as a result of compressed token construction and a larger vocabulary, the faster training resulted in a savings of more than 75 minutes over the 40 epochs. This increased performance in speed is significant since it allows for more training epochs to be completed in the same period of time.

6. CONCLUSIONS AND FUTURE WORK

The experiments in this research explored the viability of holographic reduced representations, as a means for compressing additional information into a token's fixed-dimensional vector embedding. To relate this effort to prior work for which a comparison baseline existed, an existing question answering framework was modified to evaluate the proposed methodology. The experiments done provide a launching point for future research in other deep learning frameworks, and for other natural language processing tasks.

6.1. Limitations

The expansion of the vocabulary size is the main limitation of the proposed approach. As an example, if additional information, like the output of a dependency parser, was to be holographically compressed with the token, its part-of-speech, and named-entity type, the vocabulary could become prohibitively large. This could prevent referring to stored embeddings with an index, a common practice in natural language work. Although this constraint could be circumvented by feeding the computed embeddings directly to the neural network, without first storing them in an embedding layer (that could be loaded as needed), this is undesirable for large problems as it would necessitate completion of both preprocessing and training/evaluation in single runs of the model.

6.2. Conclusions

The main contributions of this work fall in three areas: 1) construction, using HRR, of nearly orthogonal vectors representing a token's pre-trained embedding, its part-of-speech, and its named entity type, 2) documentation of results of experiments assessing the preservation of semantics after applying HRR, and 3) the favorable performance metrics attained, when compared to the baseline system. The fact that this non-optimized holographic compression model came within two percentage points of the original Document Reader, makes this a promising method for further investigation and improvements. Although one might have expected that the expansion of the size of the vocabulary by over 70% would degrade the accuracy significantly, this was not the case.

When combined with an over 30% speedup in training time, these experiments clearly demonstrate that holographically compressed token embeddings are viable in deep learning for question answering, and warrant further investigation in this and other application areas.

6.3. Future Work

This work was undertaken to set the stage for further investigation into holographically compressed token embeddings, by evaluating the necessary condition of the orthogonality of its representations, and measuring the preservation of pre-trained semantic relationships. Possible avenues for future work include:

- Characterize the semantic shifting that happens as a result of using HRR. Investigate their direct impact and identify potential remedies.
- Study the impact on performance, of compressing additional information into the token's representation. Information from a dependency parse is a good candidate.
- Remove the 4-tuple feature set from the input data. In the original work, removal of these features had very little impact on the metrics.

- Devise a method to encode sequences, such as sentences, with HRR, and use those representations as input to the network (rather than word tokens). This exploration would likely be need to be accompanied transitioning to a different (non-RNN) neural network architecture.
- Assess compressed tokens in different NLP applications. The proposed methodology was applied to question answering due to the availability of a baseline that made use of embeddings semantics and of elements that could be compressed into a single vector (part-of-speech tags and named entity typed). However, it lends itself to exploration in many other areas.

ACKNOWLEDGEMENTS

This paper was supported by Faculty Research & Creative Activity Committee (FRCAC) grant number 221751 at Middle Tennessee State University. The author is grateful for this support.

REFERENCES

- [1] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed representations of words and phrases and their compositionality", In NIPS 13, pp. 3111–3119.
- [2] Pennington, J., Socher, R., & Manning, C. D. (2014). "Glove: Global vectors for word representation". In Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543.
- [3] Plate, T.A., (1995). "Holographic reduced representations". IEEE Transactions on Neural networks, 6(3), pp.623-641.
- [4] Hochreiter, S. & Schmidhuber, J.,(1997). "Long short-term memory". Neural computation, 9(8), pp.1735-1780.
- [5] Lee, K., Salant, S., Kwiatkowski, T., Parikh, A., Das, D. and Berant, J., (2016). "Learning recurrent span representations for extractive question answering." arXiv preprint arXiv:1611.01436.
- [6] Wang, W., Yan, M. and Wu, C., (2018). "Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering". arXiv preprint arXiv:1811.11934.
- [7] Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T. and Verspoor, K., (2019). "SemEval-2017 task 3: Community question answering". arXiv preprint arXiv:1912.00730.
- [8] Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). "Reading wikipedia to answer open-domain questions". arXiv preprint arXiv:1704.00051.
- [9] Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., (2016). "SQuAD: 100,000+ questions for machine comprehension of text". arXiv preprint arXiv:1606.05250.
- [10] Jurafsky, D. & Martin, J. (2019) *Speech and Language Processing*, 3rd Edition Draft, dated October 16, 2019. Available online at <https://web.stanford.edu/~jurafsky/slp3/>. Accessed on June 11, 2020.
- [11] Tay, Y., Phan, M.C., Tuan, L.A. and Hui, S.C., (2017). "Learning to rank question answer pairs with holographic dual lstm architecture". In Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp. 695-704.
- [12] Maaten, Laurens van der, and Geoffrey Hinton. (2008). "Visualizing data using t-SNE." Journal of machine learning research 9 (Nov), pp. 2579-2605.

A Semantic Question Answering in a Restricted Smart Factory Domain Attaching to Various Data Sources

Orçun Oruç

Technische Universität Dresden, Software Technology Group,
Nöthnitzer Strasse 46, 01187, Dresden

Abstract. Industrial manufacturing has become more interconnected between smart devices such as the industry of things edge devices, tablets, manufacturing equipment, and smartphones. Smart factories have emerged and evolved with digital technologies and data science in manufacturing systems over the past few years. Smart factories make complex data enables digital manufacturing and smart supply chain management and enhanced assembly line control. Nowadays, smart factories produce a large amount of data that needs to be apprehensible by human operators and experts in decision making. However, linked data is still hard to understand and interpret for human operators, thus we need a translating system from linked data to natural language or summarization of the volume of linked data by eliminating undesired results in the linked data repository. In this study, we propose a semantic question answering in a restricted smart factory domain attaching to various data sources. In the end, we will perform qualitative and quantitative evaluation of the semantic question answering, as well as discuss findings and conclude the main points with regard to our research questions.

Keywords: Semantic Web, Web 3.0, Information Retrieval, Natural Language Processing, Industry 4.0.

1 Introduction

Currently, a vast amount of unlabeled data can not be used by applications; therefore, World Wide Web Consortium (W3C) decided to create standardization of the Web 3.0 called Semantic Web to apply Linked Open Data ¹ concept. In this concept, hypertext ad-hoc documents of the web sites have been connected through links such as Uniform Resource Identifiers (URIs)². As part of this development, Fraunhofer IWU started to organize its smart factories that are capable of generating structured linked data. Smart factories can use real-time data or linked data so as to diminish bottlenecks in assembly lines, provide predictive maintenance, enhance human-machine interaction with digitalization.

The present study introduces a human-machine-interaction concept for smart factories in terms of linked data processing integrated into a question answering.

¹ <https://lod-cloud.net/>

² <https://www.w3.org/DesignIssues/LinkedData.html>

The Semantic Web is a state-of-the-art research area that orchestrates the use of understanding in linked data between humans to machines and machines to machines. You can link data and documents to external data through linked data. In the present day, smart factories equipped with intelligent manufacturing devices, sensors, and actuators create a massive amount of data.

A semantic question answering is used for information retrieval to provide answers to questions through linked data. The proposed semantic question answering can understand complex natural language expressions, and it can respond to the user by answers. Mainly, the semantic question answering system employs unstructured data or structured data. We obtain linked data generated by an OPC-UA Server named *Dynamic Server* and the *eniLINK* [5] streaming data. The empirical analysis indicates the answer return rate and precision; therefore, it evaluates the usability for a human operator, experts, or an end-user web application. The goal of this research is to show an approach of semantic question answering for a smart factory that utilizes the natural language expressions as sentences, questions or keywords to give a precise and rapid answer to human operators or experts.

The question answering system is an essential part of human-computer interaction in the manufacturing industry. Human operators navigate a database of produced parts in the manufacturing data and the data with regard to the production line. The problem that we faced is a necessity of an aggregated information extraction tool at a smart factory by utilizing restricted domain linked data. Current researches do not tackle the problem as a whole in industrial manufacturing. We would like to solve the issue that can influence human operators or factory workers who spend a considerable amount of time on operating machines through smart devices. Question answering researchers generally perform research processes on the open-domain question answering. Even if they research restricted-domain question answering, industrial manufacturing and smart factory domain have never been observed before. Because of the amount of data size and semantically untagged streaming data in the manufacturing industry, we emphasize the importance of question answering for human operators and experts who work in different divisions in a smart factory.

The objective of this study is to develop a question answering providing preciseness and accuracy through Industry 4.0 lexicon (Uniform Resource Descriptor-based vocabulary). We would like to perform two major tasks, which are: construction of semantic triples and question answering utilizing the predefined semantic triples. The aspect of the construction of the semantic triples, question answering should use a common linked data format that is underlying semantic web technology. For instance, various data sources have different data types, which leads us to a conversion step to common linked data formats such as Resource Description Framework (RDF) or Ontology Web Language (OWL). In the context of question answering employing the predefined semantic triples, the semantic question answering systems

rely on the initiated lexicons. In the case of open-domain questions, lexicons have standards so that a developer can use them without the burden of the conversion between data formats. Due to the fact that restricted domains have no standard question answering system, we will define various benchmarking methods to find answers to our research questions.

This paper has been structured as follows: Section 2 will provide a brief overview of smart factories and Industry 4.0 and serialization process of linked data from another type of data such as the *Information Model* and from streaming data to the linked data. Section 3 introduces the theoretical background of natural language understanding and practical implementation of the question answering. Section 4 summarizes the research approach of the semantic question answering aspect of the smart factory constructed by Fraunhofer IWU. In Section 5, we implement an application and we give the implementation details of the present study. As for Section 6, we will explain the test environment; accordingly, we give the results of the semantic question answering. Section 7 explains the state-of-the-art status the *Industry 4.0* and *Smart Factories*. Then, we answer specified research questions in order to clarify key points with discussion in Section 8. Finally, we conclude in Section 9.

2 Background

2.1 Smart Factories and Industry 4.0

The definition of the smart factory has been explored over the past few years. Essentially, a smart factory consists of new integrable technological terms such as Machine Learning and Artificial Intelligence through intelligent devices such as tablets, smartphones, and sensors to make apprehensible models from unknown data areas in the manufacturing. *Industry 4.0* is a defined term that relates to the notion of smart factory bringing researchers to find state-of-the-art applications such as question answering systems, manufacturing augmented reality, and semantic sensor networks.

A smart factory is a highly digitized and connected production facility that relies on smart manufacturing [1]. This concept is one of the key outcomes of Industry 4.0, which intelligently changes manufacturing technologies. Smart manufacturing is a term coined by a set of departments of the United States [2]. The central power of the smart factory is that it makes data collection possible. Additionally, sensors enable the monitoring of specific processes throughout the factory that increases awareness of what is happening on multiple levels [3].

The development of *Industry 4.0* has a significant influence on the manufacturing industry. In the era of smart manufacturing systems, *Industry 4.0* needs to standardize all connection pipelines in smart factories. The primary objectives of

Industry 4.0 are making the manufacturing technologies of factories more capable of handling semantic triples, optimizing the chain of processes, and enhancing the capabilities of communication with each other. Moreover, *Industry 4.0* enforces end-to-end digital integration of engineering throughout the value chain to facilitate highly customized products, thus reducing internal operating costs [4].

2.2 Linked Data Serialization

The Semantic Data for OPC Unified Architecture Information Model

OPC Unified Architecture was developed for devices of industrial internet of things to remedy problems about *service orientation*, *loose coupling*, and *object-orientation paradigm*. The OPC UA has evolved from OPC to OPC UA over the past few decades and the architectural design was entirely changed. The fundamental disadvantage of OPC was that the OPC was restricting devices to connect just to Windows-based operating systems. After developing the *Distributed OPC* and *OPC UA* ideas, the foundation of *Open Platform Communications* has constructed a viable concept that consists of object-oriented, loose-coupling, and service orientation in manufacturing systems.

Aside from the OPC UA is a complex protocol; the OPC UA is one of the ubiquitous communication protocols that can be used in the various stages of the manufacturing. Thanks to the OPC client-server architecture, any devices can connect to the protocol in a manufacturing system. A programmable logic controller, a sensor, or an actuator can connect to the same server, and they can assign their values into different folder organizations to represent data in an address space. The address space is a data plane for an OPC UA server; hence it should coordinate *variables*, *methods*, *objects*, and *nodes* respectively. An end-user can identify primitive and user-defined types so that the complex structure of devices can be represented as a whole in a big data plane. However, this data plane only provides definitions and types.

The *Information Model* supports object-oriented paradigms such as abstraction and inheritance between *References* and *Objects*. It is well known that an object can live as a *Node Class* in the address space. The objects may have relationships with other objects in the *Information Model*. Utilizing *References*, a user can traverse in the address space of OPC UA to reach all levels of nodes and variables. Nevertheless, neither the *Address Space* of OPC UA nor the *Information Model* is unable to understand the meaning of data. The semantic understanding of the *OPC UA Information Model* has a vital role in performing an answering question system. The *Information Model* holds all device-specific information such as *device type*, *data changes of the device*, *vendor type*, and *relationship between devices*. These information sources can be helpful to human operators or experts aspect of system information concerning manufacturing systems.

Mapping the OPC UA Data into Semantic Data The primary data sources are semantically parsed data from *eniLINK* [5] and the OPC-UA server in Fraunhofer IWU named *Dynamic Server*. In the phase of the OPC UA server-generated data, we used an SDK, which is published by *FreeOPCUA*³ and TU-Dresden⁴. We contributed to the aforementioned projects with extra conversion steps such as XSLT and triple store processing.

OPC UA standard utilizes an information model and the information model can be used to simulate OPC UA Servers with Extensible Markup Language (XML). Due to the nature of the XML language, it is a language that depends on strong hierarchical elements and has own data model with elements and attributes that are hardly parseable. However, semantic data such as Resource Description Framework (RDF) can employ triples with the SPARQL query language.

The algorithm, as shown in Algorithm 1, identifies tree elements of a node by taking namespace indexes. The *namespace index* contains *node ids*. Once a user browses from a node to another, the user needs to know the node identification number. If the user did not scan the total number of references, the application should get all nodes that have references until the algorithm reaches all of the mesh networks. Accumulated nodes are inserted into a list to export an XML format. After obtaining XML structures, the system can convert the elements into linked data such as *Turtle RDF* through *Extensible Stylesheet Language Transformations (XSLT)*. XSLT can transform from the XML format to the RDF format by minimizing the nodes without resources called blank nodes. Once the application is converted to RDF/XML format, graph libraries can deal with the conversion process into triple formats. The application takes only care of the uniform locator identifier to do a conversion, and then the application ought to arrange uniform locators by considering different from '*example.org*'.

3 Theory of the Natural Language Understanding

In natural language processing, we need to identify the structure of a natural expression to reach the step of *Query Formulation*. The following methods that we used in the practical application are concisely given.

Preprocessing and Tokenization: Chiefly, all of the natural language processing tasks starting with preprocessing, which means cleaning the data for specific tasks that could be the reduction of non-optimized data and discrepancies between the values or removing non-related morphological properties. A question answering system should parse natural language expressions as tokens. Tokenization is the initial step for *part-of-speech tagging* to parse from a natural language to lexical grammatical structure such as verbs, nouns, cardinal numbers, or adjectives.

³ <http://freeopcua.github.io/>

⁴ https://github.com/plt-tud/opc_ua_xml_export_client

```

1: function MAINFUNCTION()                                ▷ Starting point
2:   export = ServerExport(serverurl, filename)
3:   export.IMPORT NODES(serverurl)
4:   export.EXPORT FILE(outputFile, namespaces)
5: end function
6: function BUILD NODE TREE(nodes)                       ▷ Node Formatting
7:   client ← GETENDPOINT()
8:   client ← CLIENT(serverurl)
9:   nodecumulated ← None
10:  nodeID ← 0
11:  for node < nodes do
12:    nodecumulated = node.nodeid.Namespaceindex
13:    for ref < node.getreferences() do
14:      nodecumulated.extend( ref.nodeid.Namespaceindex)
15:    end for
16:    nodecumulated = list(set(nodecumulated)           ▷ Clear duplicates
17:  end for
18:  return nodeID                                       ▷ Return node id list
19: end function
20: function IMPORT NODES(serverurl)                     ▷ Traverse Node
21:   client = Client(serverurl)
22:   client.connect()
23:   for ns < client.getNamespaces() do
24:     namespaces[client.getNamespaceIndex(ns)] = ns
25:   end for
26:   root = client.getRootNode()
27:   child = client.iterateChildNodes()
28: end function
29: function EXPORT FILE(outputFile, namespaces = None) ▷ Export into XML
30:   if namespaces != None then
31:     for node != nodes do
32:       if node.nodeid.namespaceindex is namespaces then
33:         nodes = [node]
34:       else
35:         nodes = list(nodes)
36:       end if
37:     end for
38:   end if
39:   export = XmlExport(client)
40:   export.BUILD NODE(nodes)
41:   export.appendXML(outputFile)
42: end function

```

Algorithm 1: Node Extraction

Lemmatization and Stemming: *Lemmatization* and *Stemming* are similar to each other with one difference. While a stemming algorithm is used to find syntactical structures, a lemmatization algorithm looks for a semantic structure. Stemming clears out the morphological structure of suffix and prefixes. In our proposed system, we are supposed to use a lemmatizer and stemmer to reduce lexical complexities. A lemmatizer is used to examine the morphological analysis of verbs, e.g. from "*contains*" and "*contained*" to "*contain*". Then we need to take this verb to map it into a predicate to construct a SPARQL query. The lemmatization and stemming are part of the normalization process in terms of morphological properties.

Part-of-Speech Tagging: It is a preprocessing step for parse trees to identify item taggers such as verbs, adjectives, or nouns. A sentence consists of a couple of structures including expressions like nouns, verbs, pronouns, prepositions, adverbs, conjunctions, participles, and articles that are the main categories of part-of-speech processing [7]. The part-of-speech (POS) tagger mostly applies the *Markov Model* [7] that is a part of statistical natural language understanding. The *Markov Model* stands for a state that can depend on the current state, but there is no dependency between previous states. For instance, a noun or a verb defines its neighbors, e.g. nouns are preceded by determiners, adjectives, verbs [7]. As an example, a chess player moves any chess piece according to the last movement of a rival rather than guessing from the first movement of the competitor. In this way, pre-saved corpora that have a massive amount of words have to be tagged by the POS Tagger.

Parsing: The approach of parsing is two-fold, which is the rule-based approach and the probabilistic approach [8]. The rule-based approach is a top-down approach to solve problems via predefined rules such as regex-parsing and character-based parsing. Therefore, a question answering system should define rules precisely to get the correct answer. Open-domain question answering systems use this approach because of the high complexity of the bottom-up approach and broad question types. Nevertheless, the rule-based approach could give undesirable results to question answering systems in restricted-domain so that this could be a time-wasting and an error-prone approach.

A dependency parser analyzes the grammatical structure of a sentence, and it gives information about the relationship among them. The dependency parser also defines the relationship between dependent words and root words. Thus, we can identify the center verbs or dependent nouns of complex sentences. This parser utilizes a dependency treebank file and word embedding files. Chiefly, a dependency parser applies the supervised machine learning method to reach a syntactical and semantical result. A constituency (phrase) parser is likely to be known as a phrase parser that has an objective is to check the grammatical structure of sentences by parsing the chunks of morphological structure. The constituency parser may not handle the relationship among language items. The dependency parser examines

the grammatical structure of given natural expressions to identify the relationship between a root word and dependent words that relate to the root word.

Named-Entity Recognition: It is a subtask of information extraction to locate and classify named entities with pre-classified labels, such as names of people, organizations, locations, etc. By using this method one can identify the item of a sentence as a domain-specific one. It solves the problem of recognition in the same way that the chunking method does. However, the named-entity recognition may be trained with labeled data and it is a more advanced technique than the chunking technique, which has deep and shallow parsing methods.

Similarity Analysis: Sentence similarity is used to compare two string inputs to achieve indicative questions like “*Is the system health good?*”. Mainly, this similarity method leverages averaging word vectors such as *word2vec* and *glove* that implement *Euclidean Distance*, *Manhattan Distance*, or *Cosine Similarity* [7]. In the following, three similarity methods that we analyzed are introduced:

The *Levenshtein Distance* denotes the calculation time that could be $O(|s1| \times |s2|)$ using $O(\min(|s1|, |s2|))$ space. After calculating the distance between *s1* and *s2*, the result is divided into the maximum length of string [9]. The *Jaro Winkler* has a transposition matrix *t* with common characters that are calculated together to reach the similarity value [16]. The *Jaccard Similarity* algorithm takes into consideration the size of the intersection divided by the size of the union of two sets [9]. Under the same test data and methods, similarity levels of the *Jaccard*, *Jaro Winkler*, and *Levenshtein* are 0.8095, 0.7544, and 0.58 respectively. The higher score shows a better performance for similarity measurement.

To calculate the word-based similarity, we perform the WordNet with *glove* vectors. Such vectors are pre-calculated *synset* values that are compiled and stored into a file. These *synset* values show the similarity value with the cosine similarity algorithm. The WordNet can calculate the similarity of an acronym and hypernym except for synonym. The calculation of semantic similarity is a hard and complicated process. As we will explain in the following scenario, two phrases such as ‘*Internet of Things*’ and ‘*Mesh Network*’ are semantically similar. The first implies “the network of physical objects with electronics, software, sensors, and connectivity” and the latter implies “the topology of a network whose components are all connected directly to every other component”. We cannot easily calculate this semantic similarity. Instead of calculating semantic similarity, we can calculate word vectors of verbs and nouns related to *similarity synset*. If a computed *synset* value is above the threshold value, a question answering can accept these two strings that are constructed similarly. In the practical implementation, we have used verb synonym similarity to map onto $\langle IRI: predicate \rangle$ triples.

Question Classification: Questions should be categorized to get the correct answer. It is a part of question processing that can parse the question input and assign it to the correct labels. Machine learning methods can define the derivation

of an expected answer. This paper utilized *Logistic Regression* and *Support Vector Machine* for the question classification phase. While the support vector machine was classifying the question with *TREC Dataset*⁵, the logistic regression examines the type of question at the Github repositories^{6 7}. Questions are grouped with coarse-grained labels, which are *Abbreviation*, *Entity*, *Description*, *Human*, *Location*, and *Numeric*. Furthermore, another dataset that we have trained with Logistic Regression is comprising of 'what', 'quantity questions-how many, how much, who', 'unknown', and 'why' labels. The *Logistic Regression* and *Linear Support Vector Classification* have supervised machine-learning methods by identifying coarse-grained question indicators with pre-trained labels. Logistic Regression estimates the parameter with a logistic function. The type of regression allows classifying the aforementioned labels according to multi-labels. The *Support Vector Machine* aims to improve the quality of hyperplane that separates multi-class labels. *Linear SVC*⁸ is such a method that implements a linear kernel function through the *Support Vector Machine*. The *Newton-cg* has a gradient descent function that reduces the error rate during each iteration to find out the global minimum. The *Limited BFGS* is an optimization method that can be used instead of *Newton-cg*. *Logistic Regression Cross-Validation (CV)*⁹ applies cross-validation to train and test datasets by splitting at particular percentages between them. Lastly, results have been listed in Table 1.

Parameters	Precision	F1	Recall
Newton-cg	95.55%	95.56%	95.57%
Linear SVC	92.75%	92.76%	92.77%
Limited BFGS	94.21%	94.22%	94.23%
Logistic Regression CV	95.63%	95.63%	95.64%
Linear SVC for Li-Roth Taxonomy	65%	45.5%	35%

Table 1: The evaluation of the Question Classification

4 Research Approach

Research Questions:

⁵ <https://trec.nist.gov/data.html>

⁶ <https://github.com/swapkh91/Question-Classification>

⁷ https://github.com/5hirish/adam_qas

⁸ <https://scikit-learn.org/stable/modules/svm.html>

⁹ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html

1. *RQ-1: Can a semantic question answering utilize heterogeneous linked data sources (e.g., OPC UA Information Model, streaming data, static data) in the domain of smart factory?*
2. *RQ-2: What are the requirements of the Semantic Question Answering for smart factories?*
3. *RQ-3: Can we generalize our approach to other plants and how did we contribute to the research area?*

RQ-1: Today, a smart factory creates a massive amount of data by leveraging big data analysis technology. However, the data source suffers from comprehensible by humans. This research question relates to the implementation of a serialization process into linked data. This research question evaluates the types of data sources by implementing an application.

RQ-2: This research question relates to the algorithm design thinking and domains-specific requirements to fulfill information retrieval theory and natural language understanding. This question has to evaluate the practical application.

RQ-3: This research question examines the viability of the proposal in an aspect of the division of a plant or a smart factory. Generated new test questions set to evaluate our semantic question answering.

5 Implementation

We implement a mixed parsing based approach to define essential elements of a natural query. The major priority is to detect *<subject-predicate-object>* triples and then map the verbs and nouns onto template SPARQL. This template was created according to the requirements of a smart factory. For instance, dynamic queries that fetch information from streaming data possibly need *SUM*, *AVG*, and *MIN* filter statements of SPARQL language.

As for static queries, we have hierarchical triples that contain units of the smart factory and linked data of the *Information Model*. Listing 1.1 and 1.2 show examples regarding hierarchical triples of the smart factory of eniLINK and instantiated linked data of specific domains such as OPC UA. Such predicates *<factory:contains>* should be parsed and they need to be matched with verbs. However, this may lead us to a misconception to match the synonym verb of predicates. Therefore, as illustrated in Figure 1, we inserted an extra step to identify the synonym of verbs.

Listing 1.1: Sample triples of the eniLINK hierarchical data [10]

```

1 <http://linkedfactory.iwu.fraunhofer.de/linkedfact
2 ory/linkedfactory/demofactory/machine10>
3 factory:contains
4 <http://linkedfactory.iwu.fraunhofer.de/linkedfact
5 ory/demofactory/machine10/sensor1>,

```

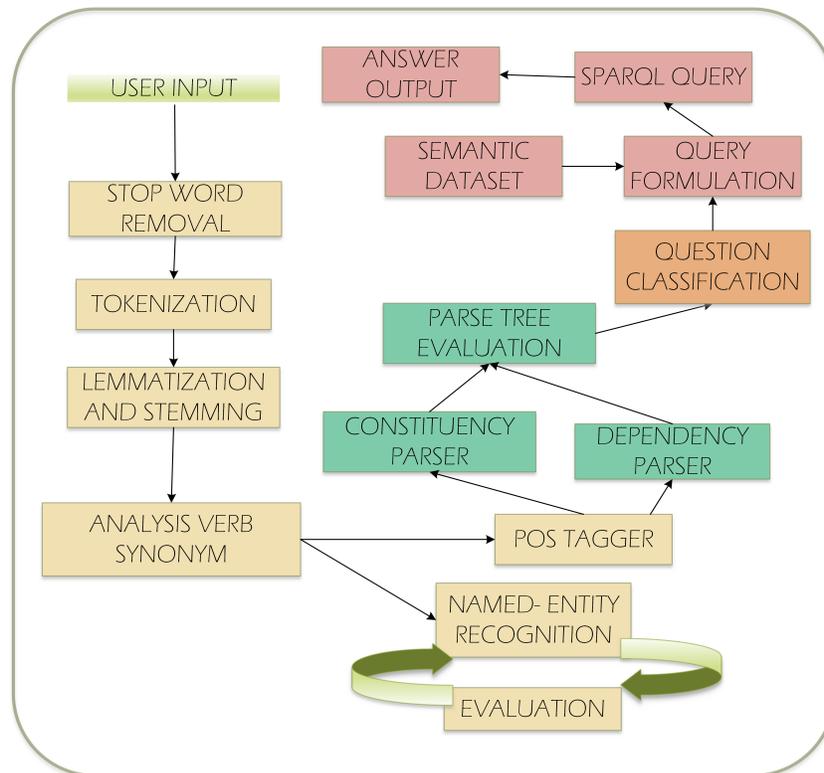


Fig. 1: Natural Language Processing for Question Answering

After taking input from any user, stop-word preprocessing starts to filter unnecessary characters such as question marks, exclamation points, commas, dots, or determiners. Tokenization is the next step to reduce the size of characters to provide optimization in natural language processing and it reduces the complexity of instances of sequence characters. Lemmatization and stemming are fundamental steps before *WordNet* verb analysis since the primary target is to extract verb, nouns, and related chunking to formulate a SPARQL query that can answer.

There is an if-else statement for the named-entity recognition after finding synonyms of the verb. As previously explained, it is a way of extracting the most common entities such as locations or names. A question answering application can face problems in identifying domain-specific names, locations, or organizations. For instance, the *linkedfactory* can be comprehensible for Fraunhofer IWU's smart factory, but another smart factory or different domain may not know what kind of entity this is. Therefore, if the question answering can catch the entity-relationship pair as shown in Figure 2, the question answering system inserts natural expressions into shallow and deep syntactic parsing.

Listing 1.2: Sample triples of the linked OPC UA Data

```

1 <unknown:namespace#UANodeSet/UAVariable_321> :BrowseName "0:
  ↪ MinSupportedSampleRate" ;
2   :DataType "Duration" ;
3   :NodeId "i=2017" ;
4   :ParentNodeId "i=2013" ;
5   :DisplayName <unknown:namespace#UANodeSet/UAVariable_321/
  ↪ DisplayName> .
6 <unknown:namespace#UANodeSet/UAVariable_321/DisplayName> rdf:value "
  ↪ MinSupportedSampleRate" .

```

For dynamic queries, the question answering system applies a similarity measurement. In Algorithm 2, the similarity flag employs a sentence similarity in the following case. *"Is the system in trouble ?"* is a reasoning query. The system should interpret this query, and the system needs to know exactly the semantic meaning of the sentence. However, the above-mentioned approach is similarity-based identification. When a user asked a question *"Is the system trouble for sensor1 in machine1?"* the semantic question answering can interpret a reasoning question through machine-readable annotations.

The architecture has provided a SPARQL endpoint for local static data, and the Key-Value Internal Service (KVIN) presents a SPARQL Endpoint for time-series data. We are using different techniques for different question types. In case of a given natural language expression as below, we can specify deep and shallow parsing diagram, as depicted in Figure 2:

"Could you tell me which one contains fofab?"

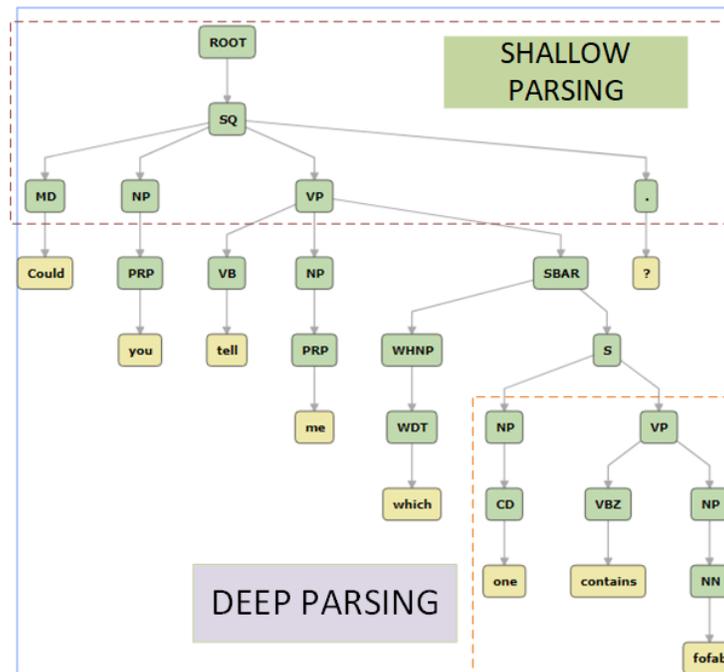


Fig. 2: An example sentence from Stanford CoreNLP [11].

We specified noun and verb phrases at a basic level so that they are using a shallow parsing that can alleviate the constituency-parsing disambiguations. If the system catches the right verb-noun pairs, it should eliminate expressions to reach the origin of the noun or verb. Such expressions may represent determiners, adjectives, or pronouns. As shown in Figure 3, the system has two verbs that it needs to map the predicate of triple onto the Turtle RDF data source. If it may find out the similarity level of *'contains'* and *'tell'*, the question answering could say the essential verb to be evaluated. However, the order of a verb is important for direct and indirect questions. As shown in Figures 3 and 4, multiple objects have relationships with the head verbs *'tell'* and *'contains'*. Subjects and objects can inverse the order of the SPARQL query. In this case, the system needs to identify universal dependencies¹⁰. The named-entity recognition can show the types of relationships as illustrated in Figure 5 and Figure 6. A drawback of this identification is a particular keyword can perplex of the identifier, noun, etc. In essence, the question answering system needs more in-depth analyses to solve the perplexities of unique keywords and open-domain words.

¹⁰ <https://universaldependencies.org/>

```

1: function QUERY FORMULATION(naturalinput)
2:   query ← QueryWithPrefixes
3:   r ← constituent.parse.tree
4:   indirectdependency ← dependendy.parse.tree
5:   while nodes ≠ leafs.terminal do                                     ▷ Until leaf nodes(Terminals)
6:     verbs ← PARSER(nodes)
7:     nouns ← PARSER(nodes)
8:     similarityflag ← WORDLATENANALYSIS(verbs)
9:     if StaticInformation is True then
10:      indirectdependencyFlag ← DEPENDENCYPARSER(nodes)
11:      if similarityflag and IndirectDependency is true then
12:        object ← nouns
13:        predicate ← verbs
14:        query += object + predicate + ?subject
15:      else
16:        subject ← nouns
17:        predicate ← verbs
18:        query += ?object + predicate + subject
19:      end if
20:    end if
21:    if DynamicInformation is True then
22:      predicate ← PARSER(nodes)
23:      object ← PARSER(nodes)
24:      similarityflag ← SENTENCESIMILARITY(input)
25:      query += object + predicate + ?subject
26:    end if
27:  end while
28:  return query
29: end function

```

Algorithm 2: Query Formulation

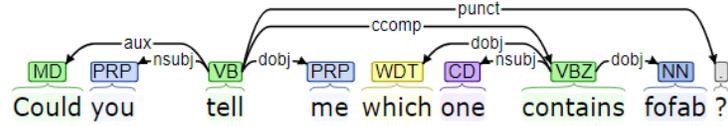


Fig. 3: A Reversed Query Dependency Parser [11]

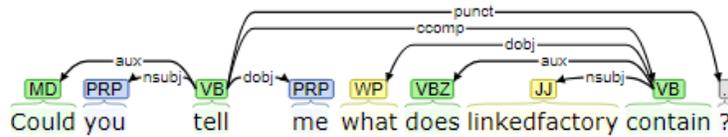


Fig. 4: A Direct Query with Dependency Parser [11]

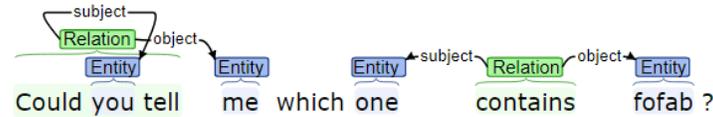


Fig. 5: Named-Entity Recognition Stanford CoreNLP [11]

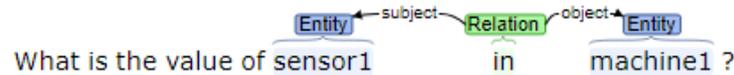


Fig. 6: Named-Entity Recognition with OpenIE [11]

6 Evaluation

6.1 Test Environment

In the evaluation phase, the data sources linked data from the OPC UA Server, eniLINK linked data that consist of elements under the linkedfactory [12] and streaming data that resides in eniLINK. As previously detailed in Linked Data Serialization (2.2), we have a heterogeneous data source for the semantic question answering. Generated data from OPC UA has no particular namespace definition

unless we define it explicitly. However, the user-defined IRIs definition has drawbacks such as collision or non-extendibility. Linked data that has been instantly generated triples makes the structure complex so that two subjects of the list can collide with identical-defined IRIs. In this case, all namespaces are generated with `http://www.example.org/` and "`<unknown_namespace>`". In Table 2, answer return rate means that an answer takes round-trip time after prompting a question or keyword in the system. Querying style indicates the type of queries that we can enter and coverage shows the source of data that has been created. As for the size parameter in Table 2, the size of the dataset that we generated from OPC UA Server has 19,687, which is 2 MB sized Turtle File. The Linkedfactory triples relate to hierarchical triples that have 70 triples as Turtle format and we test the question answering with manually generated questions through *Intel Core i7-2720QM CPU @ 2.20 GHz, 2201 MHz, and x64 based Windows 10 Pro*.

As compared to an open-domain question answering dataset, we have limited semantic triples that can be utilized by the semantic question answering. Even if the size of data is relatively big in manufacturing applications, the quality of data should be annotated and the number of predicates is one of the biggest restrictions in the semantic dataset. This restriction leads us to another restriction, which is a limited vocabulary about industrial automation.

As a result, up-to-dateness supports update statement in SPARQL in a question answering system supports. Lastly, query formulation assistance displays to the end-users about the type of assistant module that is used in a question answering system.

6.2 Result

Evaluation criteria exhibit *Recall*, *Accuracy*, *Precision*, and *F1 Score* of answers against semantic question answering system, as shown in Table 3. General evaluation parameters for a restricted domain question answering are not only limited to responding to questions but also we can assess with speed, user interaction, querying style (keywords, browsing, spell checker, abbreviation recognition). In the following formulas, TP, TN, FN, and FP denote true positive, true negative, false negative, false positive respectively.

$$Prediction = TP / (TP + FP) \quad (1)$$

$$Recall = TP / (TP + FN) \quad (2)$$

$$F1 - Score = 2x(Precision \times Recall) / (Precision + Recall) \quad (3)$$

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (4)$$

The precision (1) presents an expected answer that was correctly predicted against the total responses. F1 Score (3) is a balanced weight average between the Recall and Precision. The recall (2) is the proportion of correctly answered

questions with respect to the number of questions. The accuracy of the model (4) explains the model that has a ratio of accurately predicted observation to the entire inspection.

Test questions were created with a combination of keywords and elements of sentences, as listed in Table 4. Due to the domain restriction, the generation of test questions has a goal that responds to the questions precisely ranging from keywords to complex natural input. The target data source is a mixed source that combines static and streaming data. In the appendix, readers can observe combinations of test questions to use for further improvements.

Evaluation Parameters	Properties
Answer Return Rate	QA against generated data from OPC UA - 23.25 seconds average QA against static query from RDF file of the eniLINK - 18.92 seconds average QA against dynamic query from streaming data - 17.48 seconds QA against Template Based Open-Domain Questions - 20.55 seconds
Querying Style	Keywords-Based Search and Question-Based Search
Coverage	The eniLINK data, the linkedfactory streaming data
Size	Static data relatively small size Streaming data relatively large size
Up-to-dateness	No update statement provided by SPARQL
Query Formulation Assistance	Voice Input Recognition, Spell Checker

Table 2: The semantic question answering evaluation criterion

Question Answering Parameters	Total Questions
True Positive	34
False Negative	13
False Positive	3
Precision	94.44%
Recall	72.34%
F1 Score	81.92%
Accuracy	68.00%

Table 3: The Evaluation of the Question Answering (QA)

As for the limitation of the evaluation, manually generated test questions have been used for recall, accuracy, precision, and F1-Score. Moreover, the answer return rate is strongly dependent on system performance and web application design principles for the semantic question answering. Types of questions are mostly comprising of wh- questions and listing questions. However, restricted domain why questions (Why-Q) have been considered an irrelevant topic aspect of the semantic question answering but how questions (How-Q) are partly supported as readers can see in the Table 4.

7 Related Work

[Molla, Vicedo 2007] [13] reviewed primary characteristics of question answering in restricted domain according to integration of domain-specific information. [Molla, Vicedo 2007] [13]. defined main characteristics of question answering system over limited domains, e.g. circumscription of question answering, the complexity of question answering, and practical usage of question answering. The authors have compared between open-domain and restricted-domain question answering by figuring out key points. [Molla, Vicedo 2007] [13] offers four various aspects such as the *size of data*, *domain context*, *resources*, and *use of domain-specific resources*.

[Ferre 2012] [14] published one of the detailed reports that express common pitfalls of natural language processing and essential points while consolidating SPARQL query and morphological definitions. SQUALL is a solution for querying and updating RDF graphs by exploiting controlled natural language expressions that restrict grammar structures of a sentence to diminish complexities [14]. It has been grouped all substantial features of a morphological language, and the author pointed out what type of features in a natural language harnessed with regarding priorities and orders. The main contribution of SQUALL is categorizing ambiguities of natural expressions and how they turned an advantage out when using a controlled natural language [14].

[Biswas, Sharan, and Malik 2004] [15] proposed an architecture that extracts precise answers for a given question. The authors described the module distinctly and defined the types of questions that can be asked to the question answering. The authors sketched a translation from their intermediate language to SPARQL to gain more accuracy with their system [14]. Template-based solutions were commented on for a restricted domain and open domain question answering systems. [Unger et. al. 2012] [16] proposed a template-based solution that produces a SPARQL template, which directly matches the internal morphological features of the question.

Evaluation of a semantic question answering is still a cumbersome and hard problem. Lack of test questions that belong to a specific domain is one of the major problems. [Diekerma, Yilmazel, and D. Liddy 2004] [17] offer different methodologies from an open-domain question answering while evaluating the restricted domain question answering. The authors specify the evaluation methodology as below:

System Performance: Speed and availability **Answers:** Accuracy, Completeness **Display User Interface:** Querying styles, natural language queries, keywords, browsing, and the question formulation assistance (spell checker, abbreviation solver)

The authors stated that the *TREC* style question answering evaluation might not be suited for their restricted domain system so that user-based evaluation can be more viable to evaluate the system [17].

8 Discussion

In this chapter, we will discuss the significance of the findings relevant to the research problem being investigated. Taking our findings into consideration, we will summarize insights about problems.

First, RQ-1 and RQ-2 address distinct architectures for the use of semantic question answering. The proposal is implementing a service called KVIN that employs key-value mapping with windowed time-series data. The time-series data has been windowed with the size of data as well as the extent of the data size. Although the information structure is limited to be mapped onto Turtle triples, it can be useful for rapid prototyping. No cost will arise from designing a new language onto SPARQL or overhead of instant linked data creation from streamed data.

Generating test datasets still is a problematic topic for the restricted domain question answering systems because there could be some bias. For instance, the test dataset for the information technology domain is not valuable for a manufacturing domain, which restricts the testability; however, we have used the parameters of referenced research [17]. One of the findings is that the answer return rate is similar to template-based open-domain question answering [18]. If we want to get an answer relevant to *node id*, *node parent id*, *references*, and connected devices to OPC UA Servers, we need to convert the *Information Model of the OPC UA* to the linked data. Converting from the *root node* to the leaf nodes with namespaces of nodes would be enough to map onto $\langle \text{subject-predicate-object} \rangle$ triples. The semantic question answering should give precise answers for dynamic data and list the results of the answer against static data. Previous studies tried to solve the restricted domain question answering problem with template-based solutions by implementing a generic solution. Whereas, we perform a heuristic-based syntactic parsing to a smart factory domain. This heuristic-based approach does not guarantee optimal results in similar statements; however, it can give a high accuracy and F1-Score, as shown in Table 3.

Showing the test results of the semantic question answering and question classification, this study guides researchers of Industry 4.0 regarding how to develop an advanced dialogue system. RQ-2 defines the main features of the semantic question answering in the smart factory domain, which consists of short-listed answering,

deep-shallow parsing methods, and the use of heterogeneous data sources. The display interface may reduce the time that a human operator spends on typing and correcting spelling mistakes so that the efficiency of query processing may increase.

Consequently, as referred to RQ-3, the generalization of a semantic question answering that works in a restricted domain to another one is not an easy solution. Although the algorithm and architecture generalizations are possible; however, the drawbacks are the particular keywords in unstructured data and streaming data. Moreover, this research contributes to the research circle with algorithms regarding test set generation and features of a semantic question answering to be used against heterogeneous data sources.

The major problem of this proposal is that the question answering solely depends on the predicates of the data set defined by the smart factory. To solve the dependency problem, *subject-predicate-object* pairs can be recognized by deep learning methods with unstructured data. Correspondingly, the first finding is that the named-entity recognition had shown poor performance compared to the parsing method aspect of identifying noun and verb phrases.

The second finding is that complex paragraphs need a complicated mechanism such as co-reference resolution. Speed is another factor that we can infer when it comes to the customization of the semantic question answering. Accordingly, a technical operator or expert cannot get an answer from streaming data within the time-constraint of a mission-critical system.

The third finding is the serialization of the OPC UA can be a time-consuming task; moreover, there must be a control script to detect unaltered semantic triples. We propose the source code¹¹ so that one could recognize simulation data in OPC UA Server with a script to stave off the repercussion during serializing. The last finding is that the implementation of a generalized algorithm could degrade the precision of answers but increase the scalability at the various departments in a smart factory.

9 Conclusion

The operator assistant system increases the productivity of human operators and experts in smart factories. In this paper, we have proposed an application for a restricted domain question answering that utilizes generated data from OPC Unified Architecture and streaming data. This application can reduce the total amount of time for searching through a large number of triples. The significant findings, that are, the proposed novel approach can be used effectively to create a supervisor tool for manufacturing technologies and a synthesized human operator assistant system, which caters to a robust architecture for the aimed platform. The proposed model

¹¹ <https://github.com/zointblackbriar/QuestionAnswering>

reduces the complexity of the normalization process and employs state-of-the-art natural language understanding toolkits.

For future improvements, we plan to implement advanced semantic question answering that can be extended for time-constrained tasks such as soft-critical software systems. Furthermore, a question autocomplete system can be designed. Such a system would be efficient because it prevents the obligation of pattern or template-based question types. By scoring correctness of answers, the system can give better insight to the end-users. Lastly, named-entity recognition for smart factory and manufacturing lexicons can be added in order to eliminate a set of natural language processing steps shown in Figure 1. The more annotated data is inserted in the smart factory domain, the higher accurate reason induction, which is compatible with the question answering.

ACKNOWLEDGEMENTS

This work is funded by the German Research (DFG) within the Research Training Group Role-Based Software Infrastructures for continuous-context-sensitive Systems (GRK 1907).

References

1. R. Margaret and D. Daniel, "Definition of Smart Factory." [Online]. Available: <https://searcherp.techtarget.com/definition/smart-factory>
2. K. D. Thoben, S. A. Wiesner, and T. Wuest, ""Industrie 4.0" and smart manufacturing-a review of research issues and application examples," 2017.
3. C. Team, "What is the smart factory and its impact on manufacturing?" [Online]. Available: <https://ottomotors.com/blog/what-is-the-smart-factorymanufacturing>
4. T. D. Oesterreich and F. Teuteberg, "Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry," 2016.
5. L. D. P. IWU and F., "eniLink," 2019. [Online]. Available: <http://platform.enilink.net/>
6. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, "C-SPARQL," in *Proc. 18th Int. Conf. World wide web - WWW '09*, no. May 2014. New York, New York, USA: ACM Press, 2009, p. 1061. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1526709.1526856>
7. D. Jurafsky and J. H. Martin, *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, third edit ed., Stanford University, 2019.
8. J. Perkins, D. Chopra, and N. Hardeniya, *Natural Language Processing : Python and NLTK*. Packt Publishing, 2016.
9. P. Christen, "A comparison of personal name matching: Techniques and practical issues," in *Proc. - IEEE Int. Conf. Data Mining, ICDM*, 2006.
10. F. IWU, "eniLINK," 2020. [Online]. Available: <http://platform.enilink.net/>
11. C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 55–60. [Online]. Available: <http://aclweb.org/anthology/P14-5010>
12. F. IWU, "Linkedfactory Intro," 2018. [Online]. Available: <http://linkedfactory.iwu.fraunhofer.de/linkedfactory/view>
13. D. Mollá and J. L. Vicedo, "Question Answering in Restricted Domains: An Overview," *Comput. Linguist.*, vol. 33, no. 1, pp. 41–61, Mar. 2007.
14. S. Ferré, "SQUALL: A Controlled Natural Language for Querying and Updating RDF Graphs," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2012, pp. 11–25.
15. P. Biswas, A. Sharan, and N. Malik, "A framework for restricted domain Question Answering System," in *Proc. 2014 Int. Conf. Issues Challenges Intell. Comput. Tech. ICICT 2014*, 2014.
16. C. Unger, L. Böhmann, J. Lehmann, A. C. N. Ngomo, D. Gerber, and P. Cimiano, "Template-based question answering over RDF data," in *WWW'12 - Proc. 21st Annu. Conf. World Wide Web*, 2012.
17. A. R. Diekerma and E. D. Liddy, "Evaluation of restricted domain Question- Answering systems," *Cent. Nat. Lang. Process.*, 2004.
18. Machinalis Group, "Quepy Question Answering." [Online]. Available: <http://quepy.machinalis.com/>

Author

Orçun Oruç received M.Sc. from TU Chemnitz, and he graduated from Kocaeli University with a bachelor of science degree. Currently, he is pursuing his Ph.D. in Computer Science-Software Technology at the Dresden Technical University. His research interests include multi-agent systems, role-oriented programming, natural language processing, decentralized, and distributed applications.

Appendix

Question ID	Sample Questions	Precision	Recall
1	What do linkedfactory,heatmeter, and e3fabrik incorporate exactly ?	0.0	0.0
2	Provide me a combined result for IWU and e3sim	1.0	1.0
3	I want to know which one carries fofab ?	1.0	1.0
4	There is a member named fofab. Please give me all of its members	1.0	1.0
5	I am a customer of this company. Could you tell me please what the value of sensor1 of machine1 is ?	0.0	0.0
6	Could you tell me please what is the current value of sensor2 in machine2 ?	1.0	1.0
7	What POWERMETER holds ?	1.0	1.0
8	What does FOFAB incorporate ?	1.0	1.0
9	What does machine5 HOLD ?	1.0	1.0
10	What does gmx comprise ?	1.0	1.0
11	What comprises karobau?	1.0	1.0
12	System health for sensor2 in machine6	1.0	1.0
13	Tell me the health of system for sensor2 in machine1	0.0	0.0
14	Could you browse generated data ?	1.0	1.0
15	Give me all of the members of gmxspanen4	0.0	0.0
16	What holds coolingwater ?	1.0	1.0
17	What is the hierarchical structure of fofab ?	1.0	1.0
18	What contains IWU?	0.0	0.0
19	Could you give me the members in which contained by versuchsfeld ?	1.0	1.0
20	Could you give me the members in which linkedfactory has ?	1.0	1.0
21	What is the value of sensor1 in machine6 ?	1.0	1.0
22	What is the minimum that we can calculate for sensor1 of machine1 ?	1.0	1.0
23	What is the value of the maximum can be calculated by the sensor1 of machine1 ?	1.0	1.0
24	Could you tell me what the average for sensor3 in machine1 is ?	1.0	1.0
25	I need to learn an average value for sensor5 in machine2	0.0	0.0
26	What is the average of sensor3 in machine3 ?	1.0	1.0
27	Could you get me the references of nodes ?	1.0	1.0
28	Could you browse generated data ?	1.0	1.0
29	Is the E3-Sim member of linkedfactory ?	0.0	0.0
30	Could you take me all members of generated data ?	0.0	0.0
31	Give me all registered node id	1.0	1.0
32	I need to learn parent node id in generated data	0.5	0.5
33	Could you give me parent nodeID in the file of generated data ?	1.0	1.0
34	Give me all data blocks	1.0	1.0
35	Data blocks in generated OPC file	0.0	0.0
36	Give me the name of stations in generated data	0.0	0.0
37	All stations which are in generated data or new data	0.0	0.0
38	Registered node id	0.0	0.0
39	Who is Fofab ?	0.0	0.0
40	How is the system status for sensor1 in machine1?	1.0	1.0

Table 4: 40 Test Questions in order to test the application

A SYSTEMATIC REVIEW OF NATURAL LANGUAGE PROCESSING FOR KNOWLEDGE MANAGEMENT IN HEALTHCARE

¹Ganga Prasad Basyal, ²Bhaskar P Rimal and ¹David Zeng

¹College of Business and Information Systems,
Dakota State University, SD, USA

²The Beacom College of Computer and Cyber Sciences,
Dakota State University, SD, USA

ABSTRACT

Driven by the visions of Data Science, recent years have seen a paradigm shift in Natural Language Processing (NLP). NLP has set the milestone in text processing and proved to be the preferred choice for researchers in the healthcare domain. The objective of this paper is to identify the potential of NLP, especially, how NLP is used to support the knowledge management process in the healthcare domain, making data a critical and trusted component in improving the health outcomes. This paper provides a comprehensive survey of the state-of-the-art NLP research with a particular focus on how knowledge is created, captured, shared, and applied in the healthcare domain. Our findings suggest, first, the techniques of NLP those supporting knowledge management extraction and knowledge capture processes in healthcare. Second, we propose a conceptual model for the knowledge extraction process through NLP. Finally, we discuss a set of issues, challenges, and proposed future research areas.

KEYWORDS

Knowledge Management, Natural Language Processing, Knowledge Extraction, Knowledge Capture.

1. INTRODUCTION

Use of Natural Language Processing (NLP) has grown in the last few years, where it is preferred along with other techniques such as visualization for managing a large amount of data [1]. Natural Language Processing in healthcare is applied to various tasks such as identification of clinical relationship inpatient records [2], disease identification [3], disease classification [4] and disease prediction [5].

NLP tools are used for the prediction of disease symptoms and behaviour traits, Study by [6] suggested that NLP can be utilized for the prediction of suicide behaviour by analyzing the clinical text Notes. The author states that Clinical notes (part of Electronic health records) can be very crucial in medical informatics research which often is not utilized up to full potential. Electronic Health records contain much useful information regarding the history of patient diagnosis, treatment, prescription, healthcare status but those are in the form of text (structured and unstructured)

David C. Wyld et al. (Eds): ICAITA, CDKP, SAI, NCO, CMC, SOFT, MLT, AdNLP - 2020

pp. 275-285, 2020. CS & IT - CSCP 2020

DOI: 10.5121/csit.2020.100921

analyzing text data is very difficult and becomes more tedious in case of raw text which is highly unstructured, [2]. Information extraction from the data is of immense use as up to 18% of patient safety errors could have been reduced if proper information is available at the time of the decision-making process [7].

2. RESEARCH AGENDA

Natural Language Processing (NLP) has been utilized for knowledge management in the healthcare domain. NLP helped in gaining knowledge from a domain such as genetics under the field study of biomedical literature [8]. The author describes that genetic disorder can be related to the variation in individual genetics and can affect the medical condition or response to therapy. Gaining knowledge in the area will provide clinicians with research evidence that can be utilized for patient care.

The knowledge base in the healthcare domain is the primary source of data acquisition. These knowledge database exists in various forms, Electronic Health Records (EHR) is one of the major KM knowledge bases incorporating the patient information including patient visits, diagnosis and treatment course, diagnostic codes, patient lab reports, disease history, preventive action, clinician and nursing staff notes, etc. The advancement in EHR data management and the amalgamation of these data sources with Natural Language processing is facilitating medical information research. The data source can further be divided into two major availability sources, namely open-source and private sources. Open-source databases are those which are available for public viewing and usage. These databases can be in the form of online data provided by government agencies such as – National Health and nutrition examination Survey, the US department of agriculture data (USDA) [9]. Private sources, on the other hand, are much difficult to access as they require either permission or collaboration to access, these data sources can be in form of online information portal (i2b2 [2], patient information system), or private hospital database [10]. The Discovery of such KM data sources flourishes the common platform under which the data resides in tacit form can be converted into explicit form.

NLP tools are used for extracting the data from databases such as electronic medical records and clinical note documents [11]. Information extraction is very important in terms of obtaining complete information from raw data. Extracting information through the manually annotated document is used for identifying the phenotypic information related to inflammatory bowel disease [11]. Knowledge extraction from the large set of documents such as EHR required certain extraction tools that are capable of identifying the relationship between the text and extract the important information in form of features for higher-level understanding [12], for example extracting the particular medical terminology such as drug reaction from the whole EHR text can be done by using some of the NLP tools. A study by [13] suggests a hybrid model algorithm capable of detecting the discrepancy in medication. The author suggests an algorithm comprising of Machine learning and Natural language processing system to provide accurate detection based on gold-standards. The accuracy shown by the model in entity detection is above 95% which is promising but the detection in medication had less accuracy i.e. 65% approx. which implies that the model still needs to be tuned properly to be viable for commercial application. Discharge summaries had been used as a source of data and analysis performed on overview notes those are part of lengthy discharge notes of the pediatric patient database from Complex care medical home program based in Cincinnati children's hospital and medical center. The author used the Conditional random field (CRF) and Universal medical language system (UMLS) for the knowledge extraction system enabling feature creation, both for entity detection and medication discrepancy matching.

Knowledge gained in the domain should also be assessed in terms of quality of knowledge base created, as this is one of the major challenges faced by knowledge management systems. A study by [14] addresses this challenge pertaining to the healthcare field knowledge base by assessing the quality of psychotherapy note texts their implication focused on Post-treatment stress syndrome (PTSD). The results show that more than half of the service had been coded wrongly under psychotherapy which is eye-opening knowledge gained from research.

3. LITERATURE REVIEW

Studies related to information extraction from the EHR had been done in the past [11] where authors attempted to assess the role of the NLP system for extracting the information in the form of phenotypic data. The process is manual as it requires to create reference standard and manual annotations from clinical text notes [11]. Another study where the author attempted to embrace the medical knowledge as a key to clinical application is identified, study shows that medical knowledge updated time to time and therefore much of knowledge is stacked down in medical literature, medical knowledge is evidence-based [15] and needs empirical literature to capture information and extract knowledge out of the medical literature database.

Another study extracting the information through Electronic Health Records had been done by [16]. This study used General Architecture for text engineering (GATE) – the NLP system for knowledge extraction from Clinical records interactive search (CRIS) to evaluate the antipsychotic polypharmacy data. Results suggest that having high precision is good for the current data but the low recall is the result of a trade-off between the decision of precision and recall and High precision suggests this tool is good for polypharmacy detection from mental health EHR's.

Studies suggesting extraction of information regarding polycystic ovary syndrome [17] and influenza-like illness symptoms using a rule-based classifier [18] had been done. Research suggests that the algorithm was able to identify a problem area of clinical narrative classification as only 9.2% of all detected ILI enter in any month but future research is needed to increase the positive predictive value with another dataset.

Information extraction from patient records could provide very valuable information which is hidden and mostly untraced through the normal skimming or overview, natural language processing provides the automatic recognition of clinically important concepts and the identification of the relationship between the patients records to extract the knowledge in form of evidence which can be used for medical decision support and application, following relationship had been identified through the model proposed in [2]. The database used for information extraction was the i2b2 server and 826 patient records had been studied in research. The author states that this model could be used for information capture and discovery task in medical informatics. For information extraction stemming, stop-words elimination had been used and MetaMap is used for feature extraction under the NLP.

Information extraction from medical journal and publication is used to be of important use, a combination of the NLP and machine learning yielded very promising results in research conducted by [4]. These concepts had been extracted through the Unified medical language system (UMLS) database utilizing NLP system applications such as bag-of-words. As per the research, NLP concepts such as Biomedical concept representation is used for biomedical concept identification. The author selected a journal research article published in pub-med for extracting the information from text articles.

NLP facilitates knowledge management through the detection of disease symptoms from clinical notes [3]. These notes contain free text information from emergency department admission and discharge, details related to patient allergy, medication, previous visits had been mentioned in the clinical records. This study involves a large sample population i.e. over 40,000 and two major hospitals had participated in the study. This research was focused on the identification of common symptoms responsible for the transferability of influenza cases. The knowledge extraction process involves natural language processing system – NLP parser (pattern matching and deduction of rules) this system is capable of retrieving the information through implementing the rules created and provide the matching results.

The knowledge capture process is iteration-based, therefore, knowledge captured is always updated and changed according to the update in the domain but there is always a chance of error in the knowledge update process. This can be avoided through validation and automation of knowledge base through software process but in past EHR had been entered manually in the systems which are prone and have errors. Research conducted by [19] suggests the NLP system proposed as a negation algorithm capable of extracting symptoms reported by the patient and matching it with documented medical records.

A study by [20] aims at identifying patient exposure towards cardiovascular risk and contraindication with the help of the NLP of EMR. The author attempts to identify the aspirin usage of the patient from EMR using NLP and states that it could be a risk factor for cardiovascular disease. The proposed algorithm achieved the sensitivity of 81% compared to manual results of 84% shows that the automated process is almost equally effective as manual work. The study had limitations such as the text could only be used specifically for over the counter medication and more information is a need in part from self-report by patient, pharmacy transaction report to have accurate.

Research by [21] suggest that Natural language processing system can be used as the knowledge tool for understanding whether there is any association exists between HIV and smoking history, the association could be direct or independent. The study suggests that Less controlled HIV had fewer chances of quitting smoking even less than the chances with mental disorder to quitting smoking and Smoking found to be more prevalent in HIV-infected patients as compared to the normal sample.

The study had been done for early recognition of diseases, such as research by [22] this research provides support to KM through retrieving the 66 individual buckets (e.g., – “Paresis” had 14 different terms in ICD 9 code which then aggregated into one bucket and other like Fatigue, vision, weak, headache tingling, etc.) from Electronic health records of patients sample acquired through Columbia University Medical Center (CUMC) using natural language processing system to query based on the knowledge base of ICD-9 standards. This research is based on model-based on the model trained on the MS enriched patient dataset, the classifier was able to identify 40% of the population going to have MS before actually, it happened with AUC (0.71), sensitivity (40%) and specificity (97%). The author state one limitation as classifier was tested on local samples which is biased with the Hispanic population along with elder age and largely female patient.

Embedding the medical concepts for higher knowledge representation had been addressed in a study by [23] where the word2vec (NLP system) model is used on novel learning scheme capable of integrating the clinical notes with diagnostic code and laboratory codes. The result shows that model was successful in predicting the diagnosis codes based on the set of clinical notes and diagnosis codes and 4 Physicians testify that proposed model is far better than baseline model in predicting the

phenotypes for 3 out of 6 diseases and for the rest 3 case they had split opinion but all agreed that the proposed model had higher relevance of phenotype. The only limitation the study had was words having very less or rare frequency are not included during pre-processing. The NLP system had been adopted for modelling disease severity [24] and modelling and extracting variability [25] in using Electronic health records.

Knowledge extraction for disease identification such as methicillin-resistant staphylococcus aureus (MRSA) [26] and adverse drug reaction [27] had been implemented using the NLP systems such general architecture for text engineering (GATE) and unstructured information management architecture (UIMMA). The studies suggest that the NLP system helps in advance identification of potential ADE from EHR and implementing the NLP system can be promising for the identification of pathogen surveillance data.

The research gap is identified through studies conducted in past related to information extraction that has been related to either one disease or symptoms [11], [3] the purview of the research should be holistic and require a broad focus which encompasses higher representation of knowledge and the information extraction from NLP system supported to knowledge management in healthcare.

Identification of the NLP system as a support system for knowledge capture and extraction in most of the research had been absent thereby creating a research gap, importance of medical knowledge had been advocated in some studies such as [15] but failed to establish the relationship between knowledge management and healthcare in particular. KMS support the information flow through the extraction of information from medical databases through NLP.

4. RESEARCH FRAMEWORK

Knowledge is the highest representation of the information gained from the raw data available. The data in its basic form does not contain any direct information. Thus, tools are required to analyze the data for extracting the information out of it. This continuous flow and assimilation of information create a knowledge-base, which is the highest level in the hierarchy.

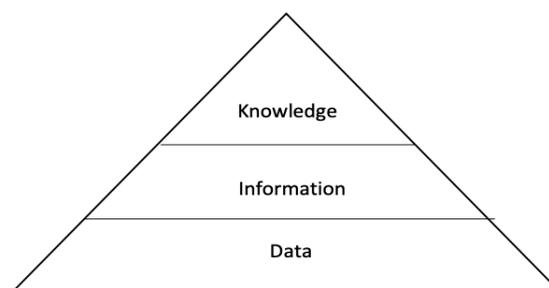


Figure 1. Hierarchy of knowledge

Knowledge is defined as the justified belief that increases an entity's capacity for effective action [28]. Knowledge management includes the KM process, which enables the smooth transition of knowledge. Two major studies in the field of knowledge management propose the four main steps. The KM process and the potential role of IT [28] incorporate four steps in the process, namely, 1) knowledge creation, 2) knowledge storage/retrieval, 3) knowledge transfer, and 4) knowledge

application. On the other hand, the KM process stated by [29] also consists of four parts: 1) knowledge discovery, 2) knowledge capture, 3) knowledge sharing, and 4) knowledge application. We adopt the KM process proposed by [28] for our research methodology.

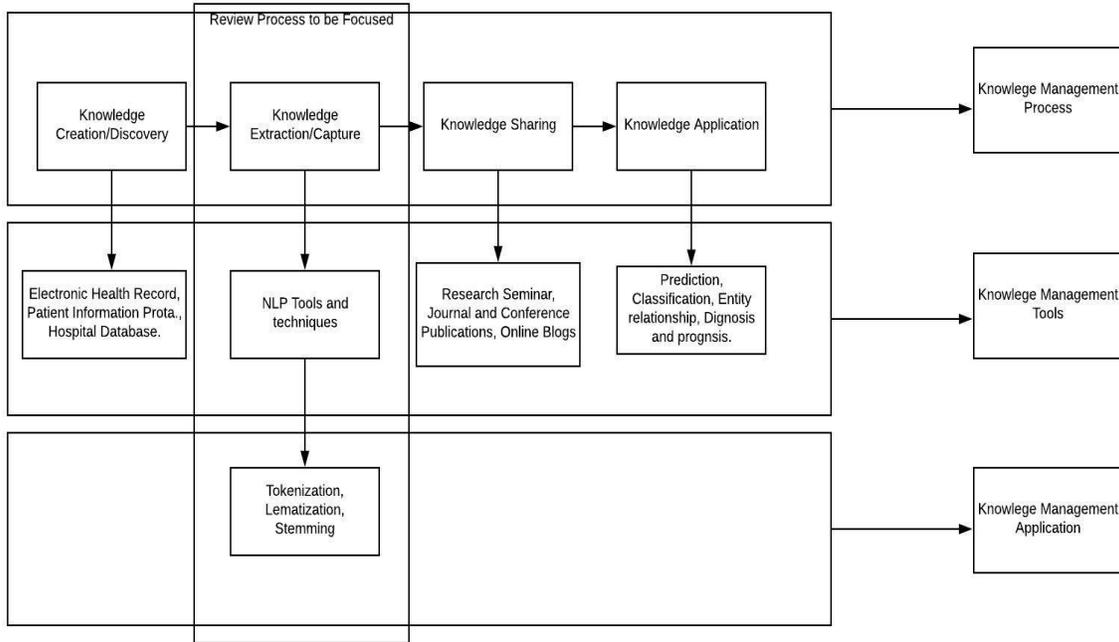


Figure 2. Research methodology framework

Our research focus is on the second process described by Alavi et al. and Beccerra et al. Both authors proposed a similar approach to the knowledge extraction one had named is as knowledge capture and others said it knowledge extraction. Knowledge created is forgotten over time, which may be due to less or no use thus this memory [28] should be retrieved before it elapses and fades away. Our review study would provide a comprehensive review of the knowledge management extraction process supported by the NLP focusing on the healthcare domain.

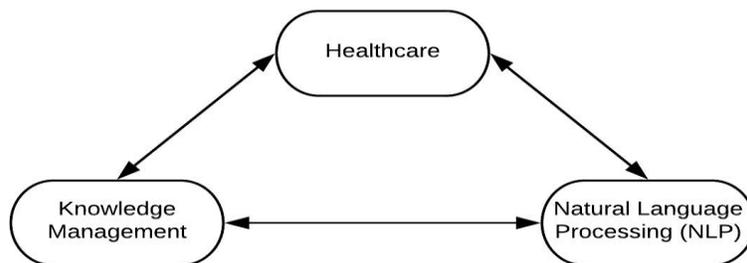


Figure 3. Knowledge flow in the healthcare domain

The NLP supported by knowledge management extracts/captures the information. The knowledge management helps in the integration of cross-domain information flow. The healthcare domain is rich in data, where information is needed for every small and big decision [7]. This information can be extracted with the help of natural language processing. Fig. 3 illustrates how knowledge management supports the NLP in extracting the information from the Healthcare domain.

For selection and screening of research article for literature review, we referred the model proposed by [30] this model had been considered as very efficient as it provides the step by step checklist for the inclusion and exclusion criteria. The model is bifurcated in four steps 1) identification, 2) screening, 3) eligibility, and 4) included.

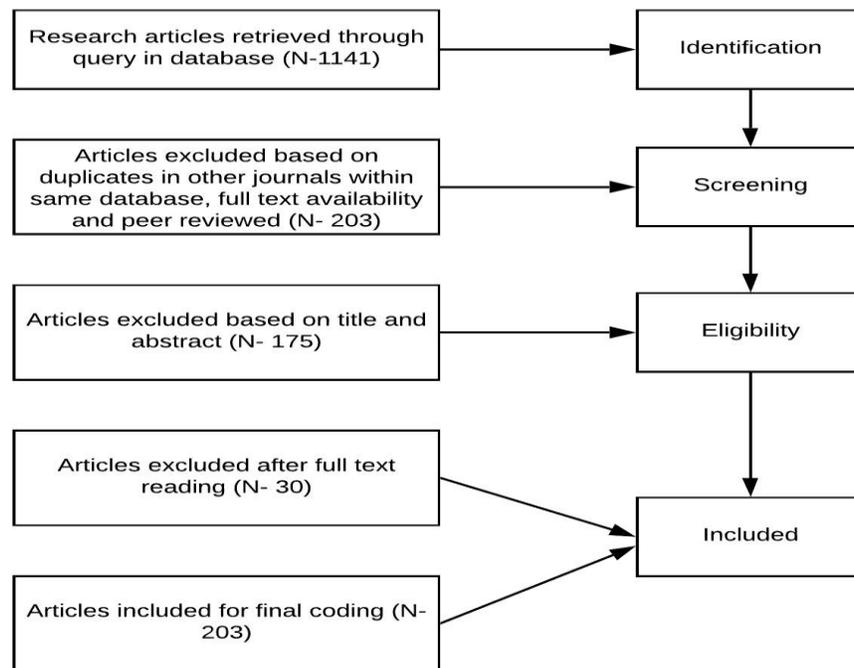


Figure 4. Article selection and screening chart ([30])

We query the EBSCO database which is considered to be one of the largest database sources for free research articles containing all major journals such as BMC, PLOS ONE, IEE, etc. During the first step, the resultant query provided us 1141 research article which we filtered further in the second step of screening. In screening, we applied 3 exclusion criteria: 1) removing duplicates, 2) peer-reviewed journal articles, and 3) full-text availability of the articles. The number of articles after the second step is 203. The third step consists of filtering articles based on the relevance of title and abstract with our research objective. The final filter applied was based on the full reading (skimming) of the article aligning with our research objective.

5. FINDINGS AND LIMITATIONS

The NLP system supports knowledge management process of information capture and extraction through specialized tools such as concept extraction based text analysis system (CETAS) for concept identification [5] and extraction in line with other application such as named entity recognition (NER) and conditional random fields (CRF) [2] had also been implemented. The NLP supports knowledge management for validation or evaluation of knowledge captured vs. knowledge documented [19]. Maker and checker in the medical information system such EHR are human and human is more vulnerable towards mistakes this problem is addressed and solved through NLP negation algorithm which compares the patient-reported symptoms with information documented.

Knowledge management process of externalization is also supported by the NLP through converting the tacit knowledge stored in medical database systems such as the unified medical language systems and converting it to explicit information [4], [13]. The NLP supports knowledge management in information capture and extraction through the NLP systems such as MedLEE proposed by [15], the general architecture for text engineering (GATE) by [16], and [27], the Word2Vec by [23], the conditional random field by [14], the Unstructured Information Management Architecture (UIMMA) [26].

We identified that majority of the research had adopted some kind of quantitative measures to compare, contrast, evaluate their research with either previously conducted similar research models or evaluating research with the same data having two splits – training and test data. [4], or to establish a new milestone with completely new methods or a hybrid model [13]. The research studies had compared their performance of two major evaluating measure – precision-recall matrix [13] and sensitivity and specificity matrix [20]. Both the matrix has a similar concept of accuracy calculation but the representing terminology is different. It depends upon the research choice which one they want to adopt.

Research studies such as [19] [13],[15], [21] suggest that the performance of the proposed model [15] had performed well providing the precision of 86% in identifying the association of phenotypes in different patient reports. 86% accuracy achieved in drug symptom and 77% for disease symptoms association and some studies even above 90% [14], [20].

We identify that many of the research studies had utilized the data to trace the prevalence of the disease [21], [11] study by Toyabe et al. provide the finding related to smoking habit stating that regular smoking percentage (33%) is less than the percentage of patients who ever smoked (47%) in life which shows that many of the patients tried and stopped the habit of smoking at some period in their life. Researches reveal that medical data is capable of disease management based on learning from the empirical or historical dataset, [23] states that their model automatically traced the diagnosis codes from the patients' EHR which can be useful for patients' prognosis and records management.

We identified that data quality is one of the barriers to achieving the set accuracy targets. The major problem with data quality is missing data. Data acquired from an open source such as online platform is low in term of quality and need much more time in cleaning and pre-processing to get acceptable quality. Another limitation is the sample size of data which tends to influence the accuracy of results [20], [24], [18], [31], [14] some of the studies even having sample sizes less than

1500 to as low as 221. This states that a limited number of observations hampers the performance of models which ultimately results in low accuracy.

6. CONCLUSION AND FUTURE AREA

Our systematic review provides a comprehensive overview of the knowledge management process, which includes knowledge extraction/capture as a sub-process supported by the NLP system focused on the healthcare domain. We conclude that knowledge management plays a very important role in the extraction of healthcare domain knowledge. This domain knowledge is further applied in the field as a knowledge system for increasing the efficiency of the designed model, which ultimately leads towards evaluating the domain challenges and serving as a decision-making tool to overcome such challenges.

Our review shows that clinical notes and patient records contain very useful and rich information but in an unstructured form which required explicit process and tools for extracting knowledge out of them. Studies had suggested ways to extract that information but had a limitation of ignoring medical data language, due to which key medical terms such as tightness, senses, anginal symptoms, and dyspnoeic had been left, resulting in low accuracy. Further research is needed in terms of model hyper tuning, which can be addressed as shown in the research framework. Future research areas could be the assessment of physician and clinical notes through information system tools in the healthcare domain.

REFERENCE

- [1] G. Albert et al., "From data to text in the Neonatal Intensive Care Unit: Using NLG technology for decision support and information management," *AI Communications*, no. 3, pp. 153–186, 2009, doi: 10.3233/AIC-2009-0453.
- [2] R. Islamaj Doğan, A. Névéol, and Z. Lu, "A context-blocks model for identifying clinical relationships in patient records," *BMC Bioinformatics*, vol. 12, no. S3, Dec. 2011, doi: 10.1186/1471-2105-12-S3-S3.
- [3] Y. Ye et al., "A study of the transferability of influenza case detection systems between two large healthcare systems," *PLOS ONE*, vol. 12, no. 4, p. e0174970, Apr. 2017, doi: 10.1371/journal.pone.0174970.
- [4] E. M. Reddy and P. Bhaskar, "Able Machine Learning Method for classifying Disease- Treatment Semantic Relations from Bio-Medical Sentences," vol. 5, no. 1, p. 5, 2018.
- [5] S. N. Hong et al., "A prediction model for advanced colorectal neoplasia in an asymptomatic screening population," *PLOS ONE*, vol. 12, no. 8, p. e0181040, Aug. 2017, doi: 10.1371/journal.pone.0181040.
- [6] C. Poulin et al., "Predicting the Risk of Suicide by Analyzing the Text of Clinical Notes," *PLoS ONE*, vol. 9, no. 1, p. e85733, Jan. 2014, doi: 10.1371/journal.pone.0085733.
- [7] D. C. Kaelber and D. W. Bates, "Health information exchange and patient safety," *Journal of Biomedical Informatics*, vol. 40, no. 6, pp. S40–S45, Dec. 2007, doi: 10.1016/j.jbi.2007.08.011.
- [8] C. Overby, P. Tarczy-Hornoch, and D. Demner-Fushman, "The potential for automated question answering in the context of genomic medicine: an assessment of existing resources and properties of answers," *BMC Bioinformatics*, vol. 10, no. Suppl 9, p. S8, 2009, doi: 10.1186/1471-2105-10-S9-S8.
- [9] K. Murakami and M. B. E. Livingstone, "Prevalence and characteristics of misreporting of energy intake in US children and adolescents: National Health and Nutrition Examination Survey (NHANES) 2003–2012," *British Journal of Nutrition*, vol. 115, no. 02, pp. 294–304, Jan. 2016, doi: 10.1017/S0007114515004304.
- [10] Q. Zhu, H. Liu, C. G. Chute, and M. Ferber, "EHR based Genetic Testing Knowledge Base (iGTKB) Development," *BMC Medical Informatics and Decision Making*, vol. 15, no. S4, Dec. 2015, doi: 10.1186/1472-6947-15-S4-S3.

- [11] B. R. South et al., "Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease," *BMC Bioinformatics*, vol. 10, no. S9, Sep. 2009, doi: 10.1186/1471-2105-10-S9-S12.
- [12] E. Iqbal et al., "ADEPt, a semantically-enriched pipeline for extracting adverse drug events from free-text electronic health records," *PLOS ONE*, vol. 12, no. 11, p. e0187121, Nov. 2017, doi: 10.1371/journal.pone.0187121.
- [13] Q. Li et al., "An end-to-end hybrid algorithm for automated medication discrepancy detection," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, Dec. 2015, doi: 10.1186/s12911-015-0160-8.
- [14] B. Shiner, L. W. D'Avolio, T. M. Nguyen, M. H. Zayed, B. V. Watts, and L. Fiore, "Automated classification of psychotherapy note text: implications for quality assessment in PTSD care: NLP of PTSD psychotherapy notes," *Journal of Evaluation in Clinical Practice*, vol. 18, no. 3, pp. 698–701, Jun. 2012, doi: 10.1111/j.1365-2753.2011.01634.x.
- [15] X. Wang, G. Hripcsak, and C. Friedman, "Characterizing environmental and phenotypic associations using information theory and electronic health records," *BMC Bioinformatics*, vol. 10, no. Suppl 9, p. S13, 2009, doi: 10.1186/1471-2105-10-S9-S13.
- [16] G. Kadra et al., "Extracting antipsychotic polypharmacy data from electronic health records: developing and evaluating a novel process," *BMC Psychiatry*, vol. 15, no. 1, Dec. 2015, doi: 10.1186/s12888-015-0557-z.
- [17] V. Castro et al., "Identification of subjects with polycystic ovary syndrome using electronic health records," *Reproductive Biology and Endocrinology*, vol. 13, no. 1, Dec. 2015, doi: 10.1186/s12958-015-0115-z.
- [18] J. MacRae et al., "Identifying influenza-like illness presentation from unstructured general practice clinical narrative using a text classifier rule-based expert system versus a clinical expert," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, Dec. 2015, doi: 10.1186/s12911-015-0201-3.
- [19] S. V. Pakhomov, S. J. Jacobsen, C. G. Chute, and V. L. Roger, "Agreement Between Patient-reported Symptoms and Their Documentation in the Medical Record," *THE AMERICAN JOURNAL OF MANAGED CARE*, vol. 14, no. 8, p. 11, 2008.
- [20] S. Pakhomov, N. Shah, P. Hanson, S. Balasubramaniam, and S. Smith, "Automated processing of electronic medical records is a reliable method of determining aspirin use in populations at risk for cardiovascular events," *Journal of Innovation in Health Informatics*, vol. 18, no. 2, pp. 125–133, Jun. 2010, doi: 10.14236/jhi.v18i2.762.
- [21] S. Toyabe, "Detecting inpatient falls by using natural language processing of electronic medical records," *BMC Health Services Research*, vol. 12, no. 1, Dec. 2012, doi: 10.1186/1472-6963-12-448.
- [22] H. S. Chase, L. R. Mitrani, G. G. Lu, and D. J. Fulgieri, "Early recognition of multiple sclerosis using natural language processing of the electronic health record," *BMC Medical Informatics and Decision Making*, vol. 17, no. 1, Dec. 2017, doi: 10.1186/s12911-017-0418-4.
- [23] T. Bai, A. K. Chanda, B. L. Egleston, and S. Vucetic, "EHR phenotyping via jointly embedding medical concepts and words into a unified vector space," *BMC Medical Informatics and Decision Making*, vol. 18, no. S4, Dec. 2018, doi: 10.1186/s12911-018-0672-0.
- [24] Z. Xia et al., "Modeling Disease Severity in Multiple Sclerosis Using Electronic Health Records," *PLoS ONE*, vol. 8, no. 11, p. e78927, Nov. 2013, doi: 10.1371/journal.pone.0078927.
- [25] G. Karystianis, T. Sheppard, W. G. Dixon, and G. Nenadic, "Modelling and extraction of variability in free-text medication prescriptions from an anonymised primary care electronic medical record research database," *BMC Medical Informatics and Decision Making*, vol. 16, no. 1, Dec. 2015, doi: 10.1186/s12911-016-0255-x.
- [26] M. Jones, S. L. DuVall, J. Spuhl, M. H. Samore, C. Nielson, and M. Rubin, "Identification of methicillin-resistant *Staphylococcus aureus* within the Nation's Veterans Affairs Medical Centers using natural language processing," *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, Dec. 2012, doi: 10.1186/1472-6947-12-34.
- [27] E. Iqbal et al., "Identification of Adverse Drug Events from Free Text Electronic Patient Records and Information in a Large Mental Health Case Register," *PLOS ONE*, vol. 10, no. 8, p. e0134208, Aug. 2015, doi: 10.1371/journal.pone.0134208.

- [28] M. Alavi and D. E. Leidner, "KNOWLEDGE MANAGEMENT AND KNOWLEDGE MANAGEMENT SYSTEMS: CONCEPTUAL FOUNDATIONS AND RESEARCH ISSUES," *MIS Quarterly*, vol. 25, no. 1, pp. 107–136, Mar. 2001.
- [29] I. Becerra-Fernandez and R. Sabherwal, *Knowledge management: systems and processes*, Second edition. Armonk, New York: M.E. Sharpe, Inc, 2015.
- [30] A. Liberati et al., "The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration," *Annals of Internal Medicine*, p. 30, Jul. 2009.
- [31] B. K. Bellows, S. L. DuVall, A. W. C. Kamauu, D. Supina, T. Babcock, and J. LaFleur, "Healthcare costs and resource utilization of patients with binge-eating disorder and eating disorder not otherwise specified in the Department of Veterans Affairs: HEALTHCARE COSTS AND UTILIZATION IN BINGE EATING," *International Journal of Eating Disorders*, vol. 48, no. 8, pp. 1082–1091, Dec. 2015, doi: 10.1002/eat.22427.

AUTHORS

Ganga Prasad Basyal is pursuing his Ph.D. in Information Systems with a specialization in decision support and analytics. He received his MBA from D.A.V.V University, India and Master in Information Systems (MSIS) from Dakota State University (DSU), Madison, SD. Currently, he is a teaching assistant at the College of Business and Information Systems, at DSU. His research area includes healthcare analytics with the inclusion of data mining and deep learning technologies.



Bhaskar Prasad Rimal (Senior Member, IEEE) is an assistant professor of Computer and Cyber Sciences at Dakota State University, Madison. He received the Ph.D. degree in telecommunications engineering from the University of Quebec, Canada. He was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque. He was a Visiting Scholar with the Department of Computer Science, Carnegie Mellon University (CMU), Pittsburgh. He was a Visiting Assistant Professor and Academic Specialist at the Department of Computer Science at Tennessee Tech University, Cookeville. He has been serving as an Associate Editor of the *IEEE Access*, Associate Technical Editor of the *IEEE Communications Magazine*, Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*, and an Editor of the *Internet Technology Letters*.



David Zeng is an assistant professor of Information Systems at Dakota State University, Madison, SD. David received his MBA and Ph.D. in MIS from the University of California, Irvine. His research focuses on the economics of IT-enabled services, the impact of IT-related factors on competition, market, and adoption of innovative services, and application of data analytic techniques in the healthcare industry. His work has been published in peer-reviewed journals and won best-paper awards in conferences. David taught undergraduate, MBA, and MS courses in China and South Korea, in areas related to management information systems, supply chain and operations management, and data analytics.



AUTHOR INDEX

<i>Adam Timmons</i>	01
<i>Alaidine Ben Ayed</i>	161
<i>Alon Schclar</i>	95
<i>Anupriya Vysala</i>	37
<i>Bhaskar P Rimal</i>	275
<i>Bolanle F. Oladejo</i>	119
<i>Chakresh Kumar Jain</i>	65
<i>Chiching Wei</i>	213
<i>Chun-Fang Huang</i>	213
<i>Darrell Schmidt</i>	01
<i>David Zeng</i>	275
<i>Derek Jacoby</i>	13
<i>Francisco Javier Cantú-Ortiz</i>	49
<i>Ganga Prasad Basyal</i>	275
<i>Gerald Onwujekwe</i>	83
<i>Gholam Aghashirin</i>	01
<i>Gopi Krishna Erabati</i>	225
<i>Hélder Araújo</i>	225
<i>Hoda S. Abdel-Aty-Zohdy</i>	01
<i>Hussain Attia</i>	189
<i>Ishan Jain</i>	131
<i>Ismail Biskri</i>	161
<i>Iva Nová</i>	201
<i>James B. Holliday</i>	107
<i>Janusz Bobulski</i>	27
<i>Jean-Guy Meunier</i>	161
<i>Jesus Leopoldo Llano</i>	49
<i>Jörg Stiller</i>	201
<i>Joseph Gomes</i>	37
<i>Kalpdrum Passi</i>	65
<i>Karel Fraňa</i>	201
<i>Kweku-Muata Osei-Bryson</i>	83
<i>Letha Etzkorn</i>	169
<i>Mariusz Kubanek</i>	27
<i>Michael Anderson</i>	169
<i>Ming-Jen Huang</i>	213
<i>Mohamed A. Zohdy</i>	01
<i>Mohammad Shojaeshafiei</i>	169
<i>Muhammad Azam</i>	13
<i>Neeraj Rattehalli</i>	131
<i>Nnatubemugo Ngwum</i>	83
<i>Nuno Gonçalves</i>	225
<i>Oladejo Olajide Ademola</i>	119
<i>Orçun Oruç</i>	251
<i>Parth Patel</i>	65
<i>Raid Saabni</i>	95
<i>Rebiha Kemcha</i>	141

<i>Salvador E. Barbosa</i>	239
<i>Samira Moussaoui</i>	141
<i>Sergio Santiago Rentería</i>	49
<i>Souhila Silmi</i>	141
<i>T.H. Ngan Le</i>	107
<i>Yvonne Coady</i>	13
<i>Zouina Doukha</i>	141