

Computer Science & Information Technology 166

Embedded Systems and Applications

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- 11th International Conference on Embedded Systems and Applications (EMSA 2022)
March 26~27, 2022, Sydney, Australia
- 11th International Conference on Software Engineering and Applications (SEA 2022)
- 8th International Conference on Artificial Intelligence and Applications (AIFU 2022)
- 3rd International Conference on Natural Language Computing and AI (NLCAI 2022)
- 3rd International Conference on Big Data and Machine Learning (BDML 2022)
- 3rd International Conference on Blockchain and Internet of Things (BIoT 2022)
- 8th International Conference on Networks & Communication (NCOM 2022)
- 11th International Conference on Cloud Computing: Services and Architecture (CLOUD 2022)
- 12th International Conference on Computer Science, Engineering and Applications (CCSEA 2022)
- 8th International Conference on Signal and Image Processing (SIPRO 2022)

Published By



AIRCC Publishing Corporation

Volume Editors

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

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

ISSN: 2231 - 5403

ISBN: 978-1-925953-65-7

DOI: 10.5121/csit.2022.120601- 10.5121/csit.2022.120628

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

11th International Conference on Embedded Systems and Applications (EMSA 2022) March 26~27, 2022, Sydney, Australia, 11th International Conference on Software Engineering and Applications (SEA 2022), 8th International Conference on Artificial Intelligence and Applications (AIFU 2022), 3rd International Conference on Natural Language Computing and AI (NLCAI 2022), 3rd International Conference on Big Data and Machine Learning (BDML 2022), 3rd International Conference on Blockchain and Internet of Things (BIoT 2022), 8th International Conference on Networks & Communication (NCOM 2022), 11th International Conference on Cloud Computing: Services and Architecture (CLOUD 2022), 12th International Conference on Computer Science, Engineering and Applications (CCSEA 2022) and 8th International Conference on Signal and Image Processing (SIPRO 2022) was collocated with 11th International Conference on Embedded Systems and Applications (EMSA 2022). 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 EMSA 2022, SEA 2022, AIFU 2022, NLCAI 2022, BDML 2022, BIoT 2022, NCOM 2022, CLOUD 2022, CCSEA 2022 and SIPRO 2022. 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, EMSA 2022, SEA 2022, AIFU 2022, NLCAI 2022, BDML 2022, BIoT 2022, NCOM 2022, CLOUD 2022, CCSEA 2022 and SIPRO 2022 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 EMSA 2022, SEA 2022, AIFU 2022, NLCAI 2022, BDML 2022, BIoT 2022, NCOM 2022, CLOUD 2022, CCSEA 2022 and SIPRO 2022.

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 (Eds)

Organization

Southeastern Louisiana University, USA
Wireilla Net Solutions, Australia

Program Committee Members

A. S. M. Sanwar Hosen,
Abdalhossein Rezai,
Abdel-Badeeh M. Salem,
Abderrahmane Ez-Zahout,
Abdessamad Belangour,
Abdullah,
Abhishek Chakraborty,
Addisson Salazar,
Adesh Kumari,
Adrian Olaru,
Afaq Ahmad,
Ahmad Yarahmadi,
Ahmed Mehaoua,
Akhil Gupta,
Akhilesh A. Wao,
Albert Bakhtizin,
Alexander Gelbukh,
Ali A Amer,
Ali Asif,
Ali Hussein Wheeb,
Alireza Valipour Baboli,
Amal Azeroual,
Amel Ourici,
Ana Luísa Varani Leal,
Anamika Ahirwar,
Anand Nayyar,
Anastasios Doulamis,
Anchit Bijalwan,
Anirban Banik,
Anita Yadav,
Annamalai Annamalai,
Anouar Abtoy,
António Abreu,
Archana Kumari,
Aridj Mohamed,
Arun Malik,
Assem Abdel Hamied Moussa,
Assia Djenouhat,
Attila Kertesz,
Auwal Salisu Yunusa,
Ayush Dogra,
Azah Kamilah Muda,
Balram Yadav,
Bin Xue,

Jeonbuk National University, South Korea
University of Science and Culture, Iran
Ain Shams University, Egypt
Mohammed V University, Morocco
University Hassan II Casablanca, Morocco
Adigrat University, Ethiopia
University of Calcutta, India
Universitat Politècnica de València, Spain
Jamia Millia Islamia Central University, India
University Politehnica of Bucharest, Romania
Sultan Qaboos University, Oman
Tarbiat Modares University, Iran
Universite Paris Descartes, France
Lovely Professional University, India
AKS University, India
Institute of the Russian Academy of Sciences, Russian
Instituto Politecnico Nacional, Mexico
Taiz University, Yemen
Harbin Engineering University, China
University of Baghdad, Iraq
University Technical and Vocational, Iran
Mohammed V University, Morocco
University Badji Mokhtar Annaba, Algeria
University of Macau, China
Jayoti Vidyapeeth Women's University, India
Duy Tan University, Viet Nam
National Technical University of Athens, Greece
Arba Minch University, Ethiopia
National Institute of Technology, India
Harcourt Butler Technical University, India
Prairie View A&M University, United States of America
Abdelmalek Essaâdi University, Morocco
ISEL, Portugal
Central University of Jammu, India
Hassiba Benbouali University Chlef Algeria
Lovely Professional University, India
Chief Eng Egyptair, Egypt
University of Algiers, Algeria
University of Szeged, Hungary
Kano State Polytechnic, Nigeria
Panjab University, India
UTeM, Malaysia
Mahakal Institute of Technology, India
National University of Defense Technology, China

Brahim Lejdel,	University of El-Oued, Algeria
Cagdas Hakan Aladag,	Hacettepe University, Turkey
Carlos Becker Westphall,	Federal University of Santa Catarina, Brazil
Chahinez Mérièm Bentaouza,	Mostaganem University, Algeria
Chandra Singh,	Sahyadri College of Engineering & Management, India
Chemesse ennehar Bencheriet,	University of Guelma, Algeria
Cherkaoui Leghris,	Hassan II university of Casablanca, Morocco
Chittineni Suneetha,	R.V.R & J.C. College of Engineering, India
Christian Mancas,	Ovidius University, Romania
Chuan-Ming Liu,	National Taipei University of Technology, Taiwan
Claude Tadonki,	MINES ParisTech, France
Claudio Schifanella,	University of Turin, Italy
Dadmehr Rahbari,	University Of Qom, Iran
Dan Wan,	Hunan Normal University, China
Dário Ferreira,	University of Beira Interior, Portugal
Dharmendra Sharma,	University of Canberra, Australia
Dinesh Reddy Vemula,	SRM University, Andhra Pradesh
Diptendu Sinha Roy,	National Institute of Technology, India
Divya Sardana,	University of Cincinnati, USA
Domenico Rotondi,	FINCONS SpA, Italy
Dongping Tian,	Baoji University of Arts and Sciences, China
El Habib Nfaoui,	Sidi Mohamed Ben Abdellah University, Morocco
EL Murabet Amina,	Abdelmalek Essaadi University, Morocco
Elzbieta Macioszek,	Silesian University of Technology, Poland
Eng Saad M. Khaleefah AL-Janabi,	Alhikma College University, Iraq
Ez-Zahout Abderrahmane,	Mohammed V University, Morocco
Fadele Ayotunde Alaba,	Federal College of Education Zaria, Nigeria
Felix J. Garcia Clemente,	University of Murcia, Spain
Fernando Zacarias Flores,	Universidad Autonoma de Puebla, Mexico
Francesco Zirilli, Sapienza	Universita di Roma , Italy
Francis Ibikunle,	Landmark University, Nigeria
G. Rajkumar,	N.M.S.S.Vellaichamy Nadar College, India
Gabriel Badescu,	University of Craiova, Romania
Gajendra Sharma,	Kathmandu University, Nepal
Ge Wu,	Southeast University, China
Giuliani Donatella,	University of Bologna, Italy
Gniewko Niedbała,	Poznan University of Life Sciences, Poland
Gopan K Shyam,	Presidency University, India
Gourav Bansal,	Kurukshetra University, India
Govindraj Chittapur,	Basaveshwar Engineering. College, India
Grigorios N. Beligiannis,	University in Patras, Greece
Grzegorz Sierpinski,	Silesian University of Technology, Poland
H V Ramakrishnan,	Dr. M.G.R. Educational And Research Institute, India
Haitham J. Taha Al-Dilleme,	University of Technology, Iraq
Hamed Taherdoost,	Hamta Group & University Canada West, Canada
Hamid Ali Abed AL-Asadi,	Iraq University college, Iraq
Hamidreza Bolhasani,	Islamic Azad University, Iran
Hanene Ben-Abdallah,	Higher Colleges of Technology, UAE
Hang Su,	Politecnico di Milano, Italy
Harisha,	Sahyadri college of engineering and Management, India
Harsha Rangrao Vyawahare,	Sant Gadge Baba Amravati University, India
Hatem Yazbek,	Nova Southeastern University, USA

Hiba Zuhair,	Al-Nahrain University, Iraq
Hilal A. Fadhil,	Solar University, Oman
Hiroimi Ban,	Sanjo City University, Japan
Hlaing Htake Khaung Tin,	University of Information Technology, Myanmar
Hosseini Rajaby Faghihi,	Michigan State University, USA
Hugo Barbosa,	Lusofona University, Portugal
Husam Suleiman,	Applied Science Private University, Jordan
Ibrahim Hamzane,	Hassan II University of Casablanca, Morocco
Ilango velchamy,	CMR Institute of Technology, India
Isa Maleki,	Islamic Azad University, Iran
Islam Atef,	Faculty of Engineering Alexandria University, Egypt
Islam Tharwat Abdel Halim,	Nile University, Egypt
Israa Shaker Tawfic,	Ministry of Science and Technology, Iraq
Israel Goytom,	Chapa Financial Technologies, Ethiopia
J. Naren,	iNurture Education Solutions Pvt Ltd, India
Jabber,	Vardhaman College of Engineering, Hyderabad, India
Jackelou Mapa,	Saint JOseph Institute of Technology, philippines
Jagadeesh HS,	A.P.S. College of Engineering, India
Jawad K. Ali,	University of Technology, Iraq
Jayavignesh T,	Vellore Institute of Technology, India
Jesuk Ko,	Universidad Mayor de San Andres, Bolivia
Jinguang Han,	Nanjing University of Finance and Economics, China
João Calado,	Instituto Superior de Engenharia de Lisboa, Portugal
Jose Silvestre Silva,	Academia Militar, Portugal
Joshila Grace,	Sathyabama Institute of Science and Technology, India
Jumana Waleed,	University of Diyala, Iraq
Jun Hu,	Harbin University of Science and Technology, China
K.L.Sudha,	Dayananda Sagar College of Engineering, India
K.V.S.S.S.S.Sairam,	NMAM Institute of Technology, India
Kamel Hussein Rahouma,	Minia University, Egypt
Kanga Koffi,	Ecole Supérieure Africaine des TIC, Cote d'Ivoire
Karim Mansour,	Salah Boubenider University, Algeria
Karima Saidi,	ICOSI Laboratory of Abbes Laghror Khenchela, Algeria
Katarzyna Zwedziak,	Opole University of Technology, Poland
Keneilwe Zuva,	University of Botswana, United Kingdom
Khurram Hameed,	Edith Cowan University, Australia
Kiran Sharma,	BML Munjal University, India
Kiran Sree,	Shri Vishnu Engineering College for Women(A), India
Kiril Alexiev,	Bulgarian Academy of Sciences, Bulgaria
Kirtikumar Patel,	Chemic Engineers USA
Klenilmar Lopes Dias,	Federal Institute of Amapa, Brazil
Kolla Bhanu Prakash,	KL University, India
Lei Meng,	Shandong University, China
Li Yan,	Xi'an Polytechnic University, China
Liquan Chen,	Southeast University, China
Ljubomir Lazic,	Belgrade UNION University, Serbia
Loc Nguyen,	Independent scholar, Vietnam
Luisa Maria Arvide Cambra,	University of Almeria, Spain
M V Ramana Murthy,	Osmania university, India
M Vijayalakshmi,	Thiagarajar College of Engineering, India
M. Khaleefah AL-Janabi,	Alhikma College University, Iraq
M. Zakaria Kurdi,	University of Lynchburg, VA, USA

M.A. Jabbar,	Vardhaman College of Engineering, India
M.K.Marichelvam,	Mepco Schlenk Engineering College, India
M.Suresh,	Kongu Engineering College, India
MA.Jabbar,	Vardhaman College of engineering, India
Maad M. Mijwil,	Baghdad College of Economic Sciences University, Iraq
Magdalena Piekutowska,	Pomeranian Univeristy in Słupsk, Poland
Mahdi Sabri,	Islamic Azad University, Iran
Mahsa Mohaghegh,	Auckland University of Technology, New Zealand
Malleswara Talla,	Concordia University, Canada
Mamata Rath,	Birla Global University, India
Mamoun Alazab,	Charles Darwin University, Australia
Manish Kumar Mishra,	University of Gondar, Ethiopia
Marcin Paprzycki,	Adam Mickiewicz University, Poland
Mario Versaci,	Univ. Mediterranea via Graziella, Italy
Masoomah Mirrashid,	Semnan University, Iran
Mayssa Frikha,	University of Sfax, Tunisia
Md.Shahjahan,	Daffodil International University, Bangladesh
Mehdi Gheisari,	Islamic Azad University, Iran
Mervat Bamiah,	Alnahj for IT Consultancy Riyadh, Saudi Arabia
Metin Soycan,	Yildiz Technical University, Turkey
Michail Kalogiannakis,	University of Crete, Greece
Mihai Horia Zaharia,	Gheorghe Asachi Technical University, Romania
Mohamed ali el sayed fahim,	benha university, Egypt
Mohamed Arezki Mellal,	M'Hamed Bougara University, Algeria
Mohamed Hamlich,	Ensam UH2C, Morocco
Mohamed-Khireddine Krolladi,	Echahid Hamma Lakhdar d'El-Oued, Algeria
Mohammad Jafarabad,	Qom University, Iran
Mohammad Reza Ghavidel Aghdam,	University of Tabriz, Iran
Mohammad Siraj,	King Saud University, Saudi Arabia
Mohammed Benyettou,	University Center of Relizane, Algeria
Mostafa EL Mallhi,	École Normale Supérieure de Fès, Morocco
M-Tahar Kechadi,	University College Dublin, Ireland
Mudasir Mohd,	University of Kashmir, India
Mueen Uddin,	Universiti Brunei Darussalam, Brunei
Muhammad Mursil,	Northeastern University, China
Muhammad Sarfraz,	Kuwait University, Kuwait
Mu-Song Chen,	Da-Yeh University, Taiwan
MV Ramana Murthy,	Osmania University, India
Nadia Abd-Alsabour,	Cairo university, Egypt
Nameer N. El-Emam,	Philadelphia University, Jordan
Narinder Singh,	Punjabi University, India
Natarajan Meghanathan,	Jackson State University, USA
Nikola Ivković,	University of Zagreb, Croatia
Nour El Houda Golea,	Batna 2 University, Algeria
Oleksii K. Tyshchenko,	University of Ostrava, Czech Republic
Omar Khadir,	Hassan II University of Casablanca, Morocco
Omid Mahdi Ebadati E,	Kharazmi University, Tehran
Otilia Manta,	Romanian American University, Romania
Ouided Sekhri,	Constantine 1 University, Algeria
P. S. Hiremath,	KLE Technological University, India
P.Gunasekaran,	Ramco Institute of Technology, India
P.V.Siva Kumar,	VNR VJIET, India

Paulo Batista,	Univresity of Évora, Portugal
Pavel Loskot,	ZJU-UIUC Institute, China
Pokkuluri Kiran Sree,	Sri Vishnu Engineering College for Women, India
Przemyslaw Falkowski-Gilsk,	Gdansk University of Technology, Poland
Quang Hung Do,	University of Transport Technology, Vietnam
Radha Raman Chandan,	Banaras Hindu University, India
Radu VasIU,	Politehnica University of Timisoara, Romania
Rajeev Kanth,	University of Turku, Finland
Rajeev Kaula,	Missouri State University, USA
Ramadan ElaieSS,	University of Benghazi, Libya
Ramgopal Kashyap,	Amity University Chhattisgarh, India
Rao Li,	University of South Carolina Aiken, USA
Rasha Thabit Mohammed,	Al-Rasheed University College, Iraq
Rinku Datta Rakshit,	Asansol Engineering College, India
S.Ganapathy	Vellore Institute of Technology, India
S.Thenmalar,	SRM Institute of Science and Technology, India
Saad M. Al-Janabi,	Alhikma College University, Iraq
Sabina Rossi,	Università Ca' Foscari Venezia, Italy
Sabyasachi Pramanik,	Haldia Institute of Technology, India
Sachin Kumar,	Kyungpook National University, South Korea
Sadaqat ur Rehman,	Namal institute-Mianwali, Pakistan
Sadique Shaikh,	AIMSR, India
Saeed Iranmanesh,	Shahid Bahonar University of Kerman, Iran
Sahil Verma,	Chandigarh University, Mohali, India
Said Nouh,	Hassan II university of Casablanca, Morocco
Samarendra Nath Sur,	Sikkim Manipal Institute of Technology, India
Samir Kumar Bandyopadhyay,	University of Calcutta, India
Samrat Kumar Dey,	Dhaka International University, Bangladesh
Sangeeta Mishra,	Thakur College of Engineering & Technology, India
Sasikumar P,	Vellore Institute of Technology, India
Sathyendra Bhat J,	St Joseph Engineering College, India
Satish Gajawada,	IIT Roorkee Alumnus, India
Sayali Kulkarni,	IIT Bombay, India
Sebastian Floercke,	University of Passau, Germany
Shahid Ali,	AGI Education Ltd, New Zealand
Shahnaz N.Shahbazova,	Azerbaijan Technical University, Baku, Azerbaijan
Shahram Babaie,	Islamic Azad University, Dubai
Shahzad Ashraf,	Hohai University, P.R China
Shamneesh Sharma,	Program Manager, upGrad Campus, India
Shashikant Patil,	NMIMS Deemed-to-be-University, India
Shervan Fekri-Ershad,	Islamic Azad university, Iran
Shi Dong,	Zhoukou Normal University, China
Shing-Tai Pan,	National University of Kaohsiung, Taiwan
Shivanand Gornale,	Rani Channamma University Belagavi, India
Shruti Bhargava Choubey,	Sreenidhi Institute of Science and Technology, India
Shufeng Li,	Communication University of China, China
Siarry Patrick, Professor,	Universite Paris-Est Creteil, France
Siddhartha Bhattacharyya,	Rajnagar Mahavidyalaya, India
Sidi Mohammed Meriah,	University of Tlemcen, Algeria
Sikandar Ali,	China University of Petroleum, China
Smain Femmam,	UHA University, France
Sofiane Bououden,	Université Abbès Laghrour, Algeria

Solomiia Fedushko,
 Sourav Sen,
 Subarna Shakya,
 Subhendu Kumar Pani,
 Suhad Faisal,
 Sujatha,
 Sumit Kalra,
 Taha Mohammed Hasan,
 Taleb zouggar souad,
 Tamara Saad Mohamed,
 Tomasz Wojciechowski,
 TV Rajini Kanth,
 Umesh Kumar Singh,
 V.Ilango,CMR
 Valerianus Hashiyana,
 Varun Jasuja,
 Varun Shukla,
 Venkata Duvvuri,
 Venkata Siva Kumar Pasupuleti,
 Vinod Kumar,
 Vyacheslav Tuzlukov,
 Waqas Haider Bangyal,
 Wei Cai,
 William Simpson,
 Xiao-Zhi Gao,
 Yanrong Lu,
 Yi Lou,
 Yingpeng Sang,
 Yousef Farhaoui,
 Yousef J. Al-Houmaily,
 Yousfi Abdellah,
 Youssef Fakir,
 Youssef Taher,
 Youye Xie,
 Yuan Tian,
 Yuansong Qiao,
 Yu-Chen Hu,
 Yuping Fan,
 Zaid Abdi Alkareem Alyasseri,
 Zewdie Mossie,
 Zhang Zhenkai,
 Zhilong Wang,
 Zhou RouGang,
 Zoran Bojkovic,

Lviv Polytechnic National University, Ukraine
 Research Scientist, USA
 Tribhuvan University, Nepal
 Krupajal Computer Academy, India
 University of Baghdad, Iraq
 Vellore Institute of Technology, India
 Indian Institute of Technology Jodhpur, India
 University of Diyala, Iraq
 Oran 2 university, Algeria
 Baghdad college of economic sciences university, Iraq
 Poznań University of Life Sciences, Poland
 SNIST, India
 Vikram University, India
 Institute of Technology, India
 University of Namibia, Namibia
 Computer Science and Engineering, India
 Pranveer Singh Institute Of Technology, India
 Oracle Corp & Purdue University, USA
 Vnr Vjiet, India
 University of Delhi, India
 Belarusian State Academy of Aviation, Belarus
 University of Gujrat, Pakistan
 Qualcomm, USA
 Institute for Defense Analyses, USA
 University of Eastern Finland, Finland
 Civil Aviation University of China, China
 Harbin Engineering University, China
 Sun Yat-sen University, China
 Moulay Ismail University, Morocco
 Institute of Public Administration, Saudi Arabia
 University Mohamed V, Morocco
 Sultan Moulay Slimane University, Morocco
 Center of Guidance and Planning, Morocco
 Colorado School of Mines, USA
 Nanjing Institute of Technology, China
 Athlone Institute of Technology, Ireland
 Providence University, Taiwan
 Illinois Institute of Technology, USA
 University of Kufa, Iraq
 Debre Markos University, Ethiopia
 Jiangsu University of Science and Technology, China
 The Pennsylvania State University, USA
 HangZhou DianZi University, China
 LSM IEEE University of Belgrade, Serbia

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)



Artificial Intelligence Community (AIC)



Soft Computing Community (SCC)



Digital Signal & Image Processing Community (DSIPC)



11th International Conference on Embedded Systems and Applications (EMSA 2022)

Mutual Inlining: An Inlining Algorithm to Reduce the Executable Size.....01-16

Yosi Ben-Asher, Nidal Faour and Ofer Shinaar

**Virtualised Ecosystem to Envisage Numerous Applications on an Automotive
Microcontroller.....**17-26

*Meghashyam Ashwathnarayan, Vaishnavi J, Ananth Kamath and
Jayakrishna Guddeti*

**Deep Learning Frameworks Evaluation for Image Classification on Resource
Constrained Devi.....**27-40

Mathieu Febvay and Ahmed Bounekkar

Leveraging OpenAmp in Embedded Mixed-Safety Critical Systems.....41-46

Mridula Prakash

**Key Learnings from Pre-Silicon Safety Compliant Bootrom Firmware
Development.....**47-54

*Chidambaram Baskaran, Pawan Nayak, R.Manoj, Sampath Shantanu and
Karuppiiah Aravindhana*

11th International Conference on Software Engineering and Applications (SEA 2022)

**Towards Maintainable Platform Software - Delivery Cost Control in
Continuous Software Development.....**55-62

Ning Luo and Yue Xiong

**Measurement of Software Development Effort Estimation Bias: Avoiding
Biased Measures of Estimation Bias.....**63-71

Magne Jørgensen

Build Automation Tools for Software Development.....73-89

Mridula Prakash

Enhanced Grey Box Fuzzing for Intel Media Driver.....91-96

Linlin Zhang and Ning Luo

8th International Conference on Artificial Intelligence and Applications (AIFU 2022)

Runway Extraction and Improved Mapping from Space Imagery.....97-108

David A. Noever

DAG-WGAN: Causal Structure Learning with Wasserstein Generative Adversarial Networks.....109-120
Hristo Petkov, Colin Hanley and Feng Dong

Application of Cross-Wavelet and Singular Value Decomposition on Covid-19 And Bio-Physical Data.....121-126
Iftikhar U. Sikder and James J. Ribero

Artificial Intelligence: Framework of Driving Triggers to Past, Present and Future Applications and Influencers of Industry Sector Adoption.....127-144
Richard Fulton, Diane Fulton and Susan Kaplan

3rd International Conference on Natural Language Computing and AI (NLCAI 2022)

An Evaluation Dataset for Legal Word Embedding: A Case Study on Chinese Codex.....145-160
Chun-Hsien Lin and Pu-Jen Cheng

An IR-based QA System for Impact of Social Determinants of Health on Covid-19.....161-181
Priyanka Addagudi and Wendy MacCaull

ESGBERT: Language Model to Help with Classification Tasks Related to Companies' Environmental, Social, and Governance Practices.....183-190
Srishti Mehra, Robert Louka, Yixun Zhang

Meeting Challenges of Modern Standard Arabic and Saudi Dialect Identification.....341-351
Yahya Aseri, Khalid Alreemy, Salem Alelyani, Mohamed Mohanna

3rd International Conference on Big Data and Machine Learning (BDML 2022)

Deep Learning Framework Mindspore and Pytorch Comparison.....191-196
Xiangyu XIA and Shaoxiang ZHOU

3rd International Conference on Blockchain and Internet of Things (BIoT 2022)

Investigating Cargo Loss in Logistics Systems using Low-Cost Impact Sensors.....197-206
Prasang Gupta, Antoinette Young and Anand Rao

Understanding the Effect of IoT Adoption on the Behavior of Firms: An Agent-based Model.....207-223
Riccardo Occa and Francesco Bertolotti

Blockchain Enabled Diabetic Patients' Data Sharing and Real Time Monitoring.....	225-235
<i>Dodo Khan, Low Tan Jung, Manzoor Ahmed Hashmani and Moke Kwai Cheong</i>	

8th International Conference on Networks & Communication (NCOM 2022)

SSL/TLS Encrypted Traffic Application Layer Protocol and Service Classification.....	237-252
<i>Kunhao Li, Bo Lang, Hongyu Liu and Shaojie Chen</i>	

11th International Conference on Cloud Computing: Services and Architecture (CLOUD 2022)

From Monolith to Microservices: Software Architecture for Autonomous UAV Infrastructure Inspection.....	253-272
<i>Lea Matlekovic and Peter Schneider-Kamp</i>	

12th International Conference on Computer Science, Engineering and Applications (CCSEA 2022)

Adaptive Forgetting, Drafting and Comprehensive Guiding: Text-to-Image Synthesis with Hierarchical Generative Adversarial Networks.....	273-285
<i>Yuting Xue, Heng Zhou, Yuxuan Ding and Xiao Shan</i>	

Trust for Big Data Usage in Cloud.....	287-304
<i>Hazirah Bee Yusof Ali and Lili Marziana Abdullah</i>	

Using Domain Knowledge for Low Resource Named Entity Recognition.....	305-316
<i>Yuan Shi</i>	

8th International Conference on Signal and Image Processing (SIPRO 2022)

An Object-Driven Collision Detection with 2D Cameras using Artificial Intelligence and Computer Vision.....	317-328
<i>Yang Liu, Evan Gunnell, Yu Sun and Hao Zheng</i>	

Unsupervised Blind Image Quality Assessment based on Multi-Feature Fusion.....	329-339
<i>Qinglin He, Chao Yang and Ping An</i>	

Mutual Inlining: An Inlining Algorithm to Reduce the executable Size

Yosi Ben-Asher
Western Digital Tefen and
The University of Haifa CS
yosi.Ben-Asher@wdc.com

Nidal Faour
Western Digital Tefen
nidal.faour@wdc.com

Ofer Shinaar
Western Digital Tefen
ofer.shinaar@wdc.com

Abstract

We consider the problem of selecting an optimized subset of inlinings (replacing a call to a function by its body) that minimize the resulting code size. Frequently, in embedded systems, the program's executable file size must fit into a small size memory. In such cases, the compiler should generate as small as possible executables. In particular, we seek to improve the code size obtained by the LLVM inliner executed with the -Oz option. One important aspect is whether or not this problem requires a global solution that considers the full span of the call graph or a local solution (as is the case with the LLVM inliner) that decides whether to apply inlining to each call separately based on the expected code-size improvement. We have implemented a global type of inlining algorithm called Mutual Inlining that selects the next call-site ($f()callsg()$) to be inline based on its global properties. The first property is the number of calls to $g()$. Next property is determining if inlining $g()$ to $f()$ may prevent inlining other more beneficial neighboring call-sites. Finally repeated inlining iterations over the call graph are performed until there are no more beneficial inlinings to perform. Hence, considering the effect of previously made inlinings on the next call-site to be inline. Our results show small but consistent improvement compare to LLVM's Oz.

1 Introduction

Frequently, in embedded systems, the program's executable file size must fit into a small size memory. In such cases, the compiler should generate as small as possible executables. There are other reasons why a smaller executable is desired, including: 1) It may run faster; 2) It can save power due to reduced number of I-cache-misses and DRAM size; 3) It can free the RAM needed for the dynamic parts of the program and other parts of the application, such as the real-time operating system. Several techniques can reduce the executable size, in particular using Overlays [1] wherein the program dynamically loads different parts of the executable, however here we consider using selective inlining (a compiler optimization) as a way to reduce the resulting executable.

Inlining [2] is a well-known compiler optimization that replaces a call statement (call-site) with the body of the called function. For example in the following code the function $g()$ (referred to as the "caller") contains two call statements to $f()$ (the "callee").

```
int A[1000],x2;
{
  while(A[t] > z) { A[t]+=z; t--; }
```

```

    return(t);
}

int g(int x, int y)
{
    x = f(x,y)+y+22;
    x += f(x,y);
    return(A[x]);
}

```

When inlining $f()$ into $g()$ the compiler may decide to replace the three calls to $f()$ by the body of $f()$ obtaining the following program.

```

int g(int x, int y)
{ int ty;
  ty=y;
  while(A[ty] > x) { A[ty]+=x; ty--; }
  x = ty+y+22;
  ty=y;
  while(A[ty] > x) { A[ty]+=x; ty--; }
  x +=ty;
  return(A[x]);
}

```

Note that in the two inlined bodies of $f()$ in $g()$ the variable x is directly used in the inlined body of $f()$ while since y is modified in the inlined body we need to save it in a temporary variable ty . This inlining affected the resulting size by the following factors:

- It saved the two call-instructions of $f()$ in $g()$ and the two return-instructions from $f()$ to $g()$.
- It possibly saved the instructions needed to pass the two parameters x, y from $g()$ to $f()$. The term “possibly” refers to the possibility that in the original program x, y have been passed by registers, not by the stack.
- It possibly saved the instructions used to save and restore some of $g()$ ’s registers according to the calling convention.
- It increased the size since the body of $f()$ is now duplicated twice instead of once in the original program.
- It increased the size due to the use of $ty = y$ compared to the original code.
- The size can also increase due to the application of optimization after the inline was applied. For example, dead-code elimination may eliminate instructions in the callee that is no longer needed due to some parameters with constant values in the call.

It thus follows that inlinings may increase the executable size mainly because the body of the callee may be duplicated in several callers. However, inlining can also reduce the executable size

as parameters passing + the calling sequence are eliminated. In addition following optimizations can reduce the size of the inlined callee even further. Inlining also affects the execution time.

We remark that inlining affects the execution time due to many other reasons (apart from the reduction in code size):

- Inlining may increase the size of the instructions-cache, creating in-cache-copies of the same function competing with each other.
- Inlining can improve the scheduling increasing the ILP.
- Inlining can allow us to apply optimizations such as CSE and invariant code motion across functions if they happen to be inlined inside the same function.

Previous works on inlining techniques considered a different problem than the one addressed here, namely that of finding the subset of inlinings that increase performance the most while not exceeding a given budget of memory size. This memory budget refers to the executable size that is obtained when no inline is applied. Typically these works present a heuristic that balances the expected increase in performance due to a given inline with the expected size increase due to that inline. The execution frequency of that call usually estimates the expected performance due inline a given call. This differs from the problem considered here, namely to select the best subset of inlinings (caller-callee pairs) that minimize the resulting executable size the most.

The proposed algorithm computes such a subset of profitable call-sites (caller+callee). Unlike the LLVM's Oz inliner that scans the call graph and separately decides whether or not to inline a given call, the proposed algorithm considers a more global type of solution wherein the effect of inlining a call on its neighboring calls is taking into account. This is because inlining one profitable call site may cause an even more profitable neighboring call site to become non-profitable. As such, the proposed algorithm selects a better subset of inlinings comparing to the LLVM's -Oz inliner.

Section 2 contains a formal graph based model for evaluating the effect of inlining. Next, section 3 specifies the proposed inlining algorithm, while section 4 compares the main features of the proposed algorithm to those of the regular -Oz inliner. Finally, section 5 compares the executable's size of the Oz-inliner vs. the proposed MI-inliner for a selected set of C/C++ programs.

2 Formalizing the problem

Here we describe a simplified model for the inlining problem of finding a subset of call sites from a given call graph that minimizes the size of the resulting executable. The model contains the call graph wherein each edge indicate a call; it is a simplification of the real situation. Thus with every part of the model, we indicate in what way it is a simplification:

- We are given a call graph G (directed a acyclic graph) whose nodes are functions f_1, \dots, f_n and an edge $f_i \longrightarrow f_j$ indicates a call to f_i from f_j .

The simplification part is that we assume that all the calls of the program are given as edges. This ignores the fact that some calls in a program will not be specified as edges in G as:

- There are call-sites that call a function via a pointer (indirect calls); hence, the callee is unknown, and no edge for this call will be produced in G .

- G is, in fact, the call graph per module as usually, applications contain multiple modules that are compiled separately. Hence, if the callee is an external function, its body may not be available, and G will not include an edge for this call.
- Each node f_i has a size $|f_i|$ indicating the size in bytes of f_i 's code. If we do not apply any inline the cost (total size of the executable) is $S = \sum_{i=0}^n |f_i|$.

This is a simplification as at the compilation stage in which the inlining is performed The function's code is given in its Intermediate representation (IR instruction) form [3] not in the final machine code, which will be generated after several stages of optimizations and machine-code generation. The functions' size we get are estimations of the IR instructions that will be eliminated by the following optimizations and to the number of machine instructions will be finally generated.

- Each edge $f_i \longrightarrow f_j$ of G is labeled by a size reduction e_i indicating the reduction in the size of f_j if we will apply the inline of f_i into f_j . This size corresponds to:
 - The elimination of the calling sequence (call+return instructions).
 - Elimination of the callee's instructions for saving/restoring the caller's registers, as now the callee's body becomes a natural part of the caller's code.
 - Elimination of parameter's passing in the caller, as after inlining the callee's instructions will directly access the caller's registers.

This holds if we ignore further possible reduction in the size of $f_j + f_i$ caused by optimizations such as dead-code elimination and function specialization applied after inlining. Also, this does not account for a possible increase in the register pressure caused by adding the callee's body to the caller (possibly adding more spills to the caller).

- we assume that all the incoming edges (calls) to a function f_i has the same saving e_i . since the saving due to inlining depends on the call to f_i which is the same for every f calling f_i . This is a simplification since different calls to the same function may have other savings, as the saving of each call depends varies due to the number of parameters set to constants in every call. This assumption is only made for simplification in the description, but in the proposed algorithm's actual realization, we compute the estimation to each call separately.
- The sum of the e_i s of outgoing edges of f_j (all the calls made by f_j) must be smaller than the size of f_j so that the result of inlining all the calls of f_j will not produce negative numbers. Thus inlining may reduce the size of f_j but never eliminate it.

Thus if we inline $f_i \longrightarrow f_j$ then

- The size of f_j becomes $|f_j| + |f_i| - e_i$.
- The edge $f_i \xrightarrow{e_i} f_j$ is eliminated.
- For every $f_k \xrightarrow{e_k} f_i$ (function called by f_i) we need to add an edge $f_k \xrightarrow{e_k} f_j$ to G .
- If the number of calls to f_i is zero it is eliminated from the graph.

Figure 1 depicts two inlining steps and their effect on the resulting size. Note that for inlining a node with *outdegree* = 1 is always beneficial, $S = S - e3$ after inlining $f3$ into $f4$, however the second inlining $f4 + f3$ into $f5$ ($f3 + f4$ has *outdegree* = 2) add the cost $S = S + |f4| + |f3| - e3 - e4$. Thus we need that $|f4| + |f3| - e3 - e4 \leq 0$ in order for this inlining to be profitable.

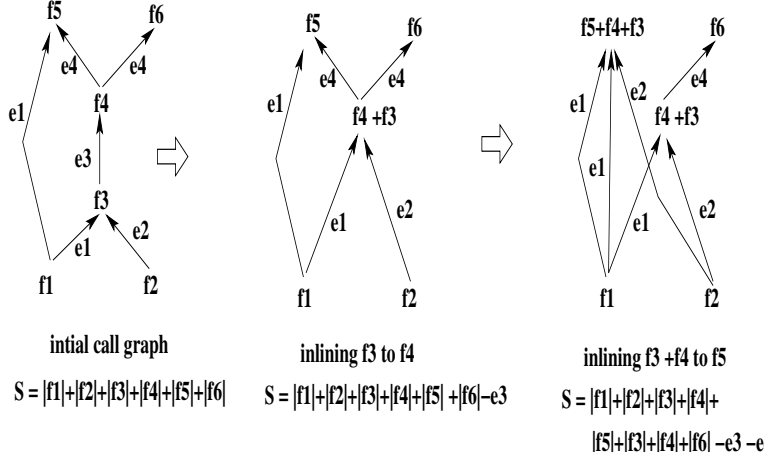


Figure 1: Sequence of inline steps and the resulting size.

Defining inlining this way allow us to specify the problem as follows:

Definition 2.1 Given a call graph G^0 such that

- The nodes of G^0 are functions f_1, f_2, \dots and the size of each f_i ($|f_i|$) is attached.
- The edges of G^0 $f_i \xrightarrow{e_i} f_j$ are labeled by the saving $f_i \xrightarrow{e_i} f_j$ that will be obtain after inlining the call $f_i \xrightarrow{e_i} f_j$.

The goal is to find a sequence of inlinings:

$$G^1 = \text{inline}(f_i \xrightarrow{e} f_j \in G^0), G^2 = \text{inline}(f_k \xrightarrow{e} f_m \in G^1), \dots \dots G^k = \text{inline}(f_z \xrightarrow{e} f_r \in G^{k-1}),$$

for which The size $S = \sum_{f_i \in G^k} |f_i|$ is the smallest overall possible subsets of inlinings (excluding recursive calls).

Note that there are finite number of possible subsets hence this definition is valid as after each inline the number of edges in the resulting $G^t = \text{inline}(f_z \xrightarrow{e} f_r \in G^{t-1})$ is smaller than the edges in G^{t-1} .

3 Proposed Algorithm

LLVM inlining algorithm works by inlining G 's nodes bottom-up in topological order. All its call-sites (calls to it from other functions) are examined to see if they are profitable for each node. A call site (edge) $f_i \xrightarrow{e} f_j$ is profitable if $|f_j| - e \leq \text{threshod}$. For the Oz option, this threshold = 5, basically meaning that the overall size is reduced due to this inlining. However, this bottom-up

order may not always be the correct one (i.e., obtain the optimal result). Figure 2 depicts a case wherein by selecting different values to e_i and $|f_i|$ it may be better to do either the upper inlining (inlining-1) or the bottom inlining (inlining-2). If $|f_3| - 2e_3 < -e_1 - e_2$ then inlining-1 is preferable otherwise inlining-2 (LLVM style) is preferable. Also as can be seen eventually we get the same cost which, compare to the original cost S , is now $S + |f_1| + |f_2| + |f_3| - 2 \cdot e_1 - 2 \cdot e_2 - 2 \cdot e_3$. This may not be beneficial compare to doing just one inlining. This proves that applying LLVM inlining (bottom-up) may not always be beneficial.

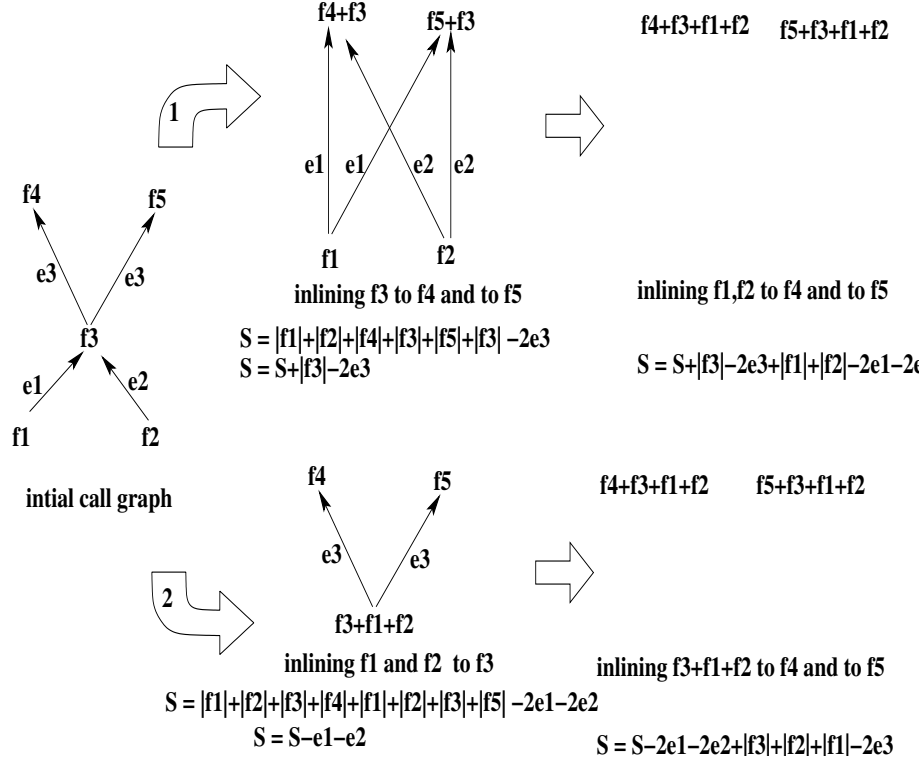


Figure 2: Depend on the values of e_i and $|f_i|$ different inlining should be selected.

Next we differentiate between two types of nodes:

Tree nodes- Nodes/functions with a single, i.e. functions with only one call-site. Tree nodes should always be inlined so that maximal tree-like sub-graphs are always inlined to the root caller. This is a safe move as, after inlining, the body of the callee does not remain in the executable, and size reduction is guaranteed. Hence inlining a node with in-degree one is always part of the optimal subset of selected inlinings.

Star nodes- Nodes/functions with more than one caller. A profitable star is a node that inlining it to all its callers will decrease the overall code size. Given $outdegree = d$ and edges $f_i \xrightarrow{e_i} f$ $i = 1 \dots d$ a profitable star satisfies that $\sum_{i=1}^d e_i + |f| \geq \sum_{i=1}^d |f_i|$.

Note that we have defined a profitable inline of a star as inlining of all the calls to this node. Thus we excluded the possibility that inlining some (but not all) calls to this node can be profitable.

This is based on the claim that if it is not profitable to inline all the node's outgoing edges, then inlining a partial subset of its edges can not be a part of an optimal solution (one that achieves the minimal size). The validity of this claim is due to the additive linear combination of the overall size and the fact that each outgoing edge of a node contributes the same saving to the final cost. Figure 3 is an example showing that partial inlining of a star's calls is not profitable. The initial G has seven nodes and a star at $f1$ with three calls ($f2 \rightarrow f1$, $f3 \rightarrow f1$, $f4 \rightarrow f1$). The initial cost (size) is 206. The figure depicts three inlinings of the $f1$ -star (left to right):

- Inlining only $f4 \rightarrow f1$ and then completing the remaining inlining of tree nodes $f5 \rightarrow f4 + f1$ and $f6 \rightarrow f5 + f4 + f1$. This reduce the overall size to 196.
- Full inlining $f4 \rightarrow f1, f3 \rightarrow f1, f2 \rightarrow f1$ and then completing the remaining inlining of tree nodes $f5 \rightarrow f4 + f1$ and $f6 \rightarrow f5 + f4 + f1$. This increase the overall size to 236. and then completing the remaining inlining of tree nodes $f5 \rightarrow f4 + f1$ and $f6 \rightarrow f5 + f4 + f1$.
- Inlining none of the calls to $f1$ and then completing the remaining inlining of tree nodes $f5 \rightarrow f4$ and $f6 \rightarrow f5 + f4$. Indeed this option obtains the best score reducing the overall size to 186 which is better then what is obtained by the partial inlining $f4 \rightarrow f1$.

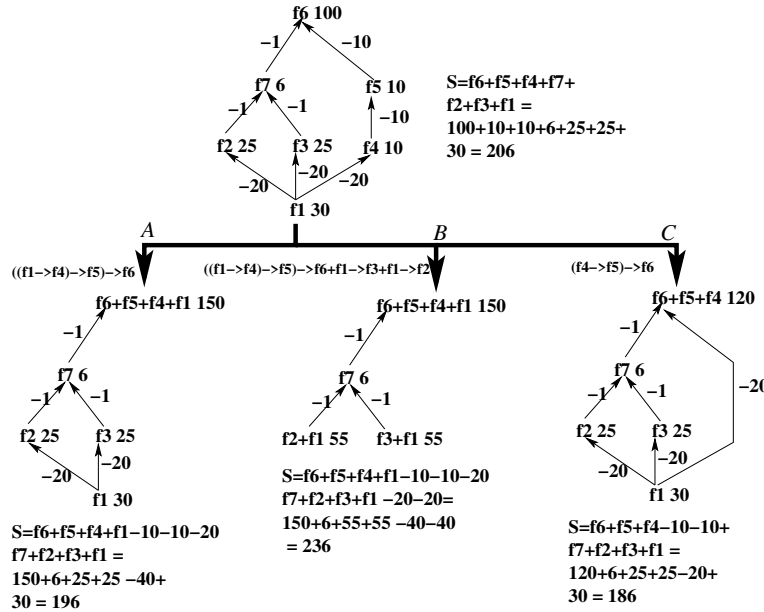


Figure 3: *Partial inlining of a star's calls is not beneficial.*

This could have lead to the following greedy algorithm:

1. Inline all the tree nodes until we remain with only star-nodes.
2. Compute the profit of the star nodes and inline all profitable stars in the order of their expected profit (i.e., inline the most profitable nodes first).
3. Repeat these two steps until there are no more sub-trees and profitable starts to inline.

The need to inline profitable stars according to their expected profit is illustrated in figure 4. Initially there are two profitable stars: one at node $f5$ and one star at node $f2$. The star at node $f5$ yields a profit of $f + 7 - 6 - 6 = -5$ and the star at $f2$ yields a smaller profit of $10 - 6 - 6 = -2$. Inlining all other stars ($star - f1$, $star - f4$, $star - f3$) are not profitable since they have a positive cost, e.g. ($cost(star - f4) = 10 - 3 - 3 > 0$). Indeed starting with inlining the star at $f5$ yields a better size of 72.

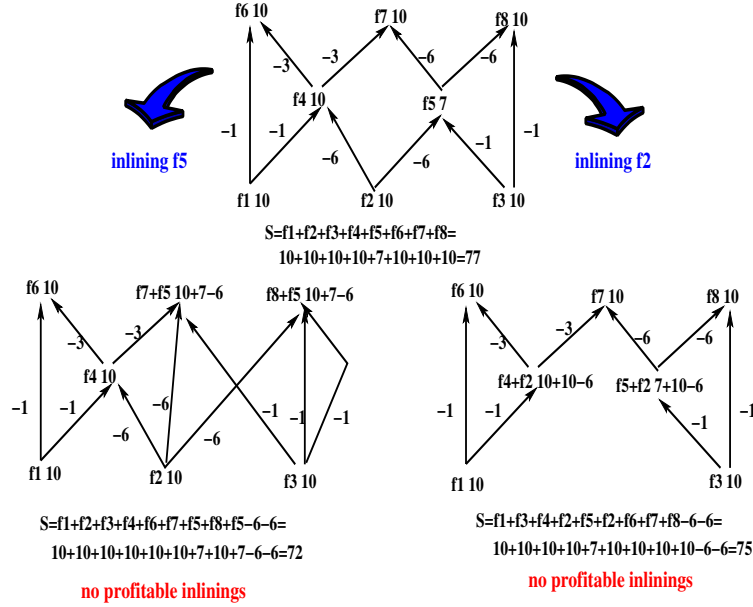
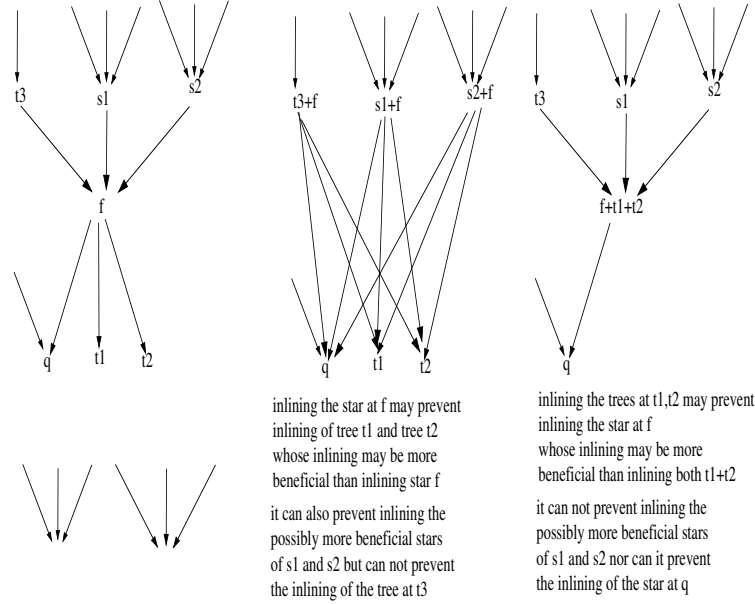


Figure 4: *Inlining the most profitable star.*

Figure 2 (the one used earlier to show that bottom-up order may be wrong) also shows that inlining trees first is not optimal. This is because we can set the numbers $(e_i, |f|)$ such after inlining $f3 \rightarrow f2$, $f3 \rightarrow f1$ the inlining of the star at $f3$ is no longer profitable. Recall that the figure depicts a case wherein by selecting different values to e_i and $|f_i|$ it may be better to either perform the upper inline (inlining-1) first or the bottom inlining (inlining-2). If $|f_3| - 2e_3 < -e_1 - e_2$ then inlining-1 is preferable otherwise inlining-2 is preferable. This reveals another more general problem, namely the “mutual effect” wherein inlining a node f can affect the profitability of inlining f ’s neighboring nodes (all its callers and its callee nodes). In figure 2 it can happen that after inlining $f3 \rightarrow f1$ the neighboring star at $f3$ is now at size $|f3 + f1 - e_1|$ which may become too “heavy” for inlining. Also, the numbers can be set such that if we inline the $f3$ -star first, we transform the two tree nodes $f1, f2$ to stars (as is depicted by inlining-1). Now it can happen that originally the two tree-inlinings at $f1$ and $f2$ are more profitable than the inline of the $f3$ -star; however, inlining $f3$ first will prevent us from inlining the more profitable $f1, f2$ inlinings. Figure 5 depicts the mutual effect of inlining a star-node on inlining its neighboring tree/star-nodes and the opposite effect of inlining a tree node on its neighboring star nodes. Clearly, inlining a tree node can not affect the profitability of inlining its neighboring tree nodes.

Figure 5: *Different cases of the mutual inlining effect.*

We can simplify the number of cases of the mutual effect by assuming that

- a star node $s1$ always prevents the inlining (conflicts) of all its more profitable parent star-nodes (callers).
- a star node $s1$ always prevents the inlining (conflicts) of all its more profitable sons tree (out-degree = 1) nodes (the callees).
- a tree node always prevents the inlining (conflicts) of all its more profitable father star nodes.

This leaves us with only two mutual-effect cases, i.e., conflict cases: A) between two profitable neighboring star-nodes and B) between a profitable tree-node and a profitable star-node.

Given the initial graph G^0 where

- The nodes of G^0 are functions f_1, f_2, \dots and the size of each f_i ($|f_i|$ is attached).
- The edges of G^0 $f_i \rightarrow f_j$ are labeled by the saving $f_i \xrightarrow{e_i} f_j$ that will be obtain after inlining the call $f_i \rightarrow f_j$.

the proposed inlining algorithm that consider the mutual inlining effect (called the MI algorithm) works as follows:

1. $t = 0$;
2. Select all the profitable nodes (star/tree-nodes) in G^t . If no profitable nodes are found, then exit the algorithm.
3. Compute a conflict graph CG whose nodes are all the profitable nodes of G^t and its edges (non-directed) correspond to conflicts between two neighboring nodes of CG .

4. Compute the Maximum Independent Set ($MXIS_{G^t}$) of CG [4]. Since Computing MXIS is np-hard we are using a greedy approximation algorithm. This greedy algorithm forms a maximum independent set by, at each step, choosing the next node f in CG with the highest $\frac{profit(v)}{degree(v)}$ and removing its neighbors.
5. The nodes of $MXIS(G^t)$ are inlined one after the other in the order they were added to $MXIS(G^t)$. This inlining of $MXIS(G^t)$ nodes forms a newly updated graph G^{t+1} as defined by the inlining operation.
6. A limited set of optimizations is applied on the nodes of G^{t+1} .
7. The saving on the edges and the size $|f|$ on each node are updated as well. For example, inlining $f_3 \xrightarrow{e_3} f_1$ will result in a new node $f_3 + f_1$ with new weight $|f_3| + |f_1| - e_3$. In addition the saving of each of $f_3 + f_1$ callers will be recomputed to adapt to the new modified body of $f_3 + f_1$ as now more instruction may be eliminated in $f_3 + f_1$ body due constants passed as parameters in $f_3 + f_1$ caller. Changes in $|f_i|$ and e_i of G^{t+1} are also affected by the optimizations we have applied.
8. Iterate these steps until no more profitable nodes are found.

4 Comparing MI with LLVM's -Oz

Here, we compare the main features of the Mutual Inlining Algorithm (MI) to that of the LLVM Inliner with the -Oz option (Oz). We compare the inlining decisions of LLVM Oz vs. those of the MI algorithm. We use small examples and use actual printings of the call-graph G made by the LLVM's inliner (Oz/MI) during compilation (in dot format).

Inline Cost estimation - Both algorithms use the following measurements:

- function size (FS), namely number of instructions in the function body where the Oz counts the number of the IRs, and the MI counts the expected size in bytes of the IRs (using the expected size after code generation). This is done via unique conversion of the IR instructions to the expected RISC-V instructions (in bytes). This conversion was added to LLVM's inlineCost.cpp analyzer.
- Simplifies instructions (SI). For a given CS (call-site) SI counts the remaining instructions in $g()$'s body after eliminating instructions related to constant values passed to the callee at the call-site. Again Oz uses IRs; MI uses the byte size of IRs.
- Saving of inlining a CS due to elimination of parameters' passing and the call/return sequence (again Oz counts IRs NI counts byte-size of IRs).

Calculating the effectiveness of inlining a given CS - The Oz is pessimistic; it assumes that the body of the caller will remain either due to external calls from other modules or due to the possibility of an additional indirect call to g via a function-pointer which will not be inlined. The Oz, therefore, apply the inlining only if

$$SI - (Saving + staticbonus) < Threshold$$

where a static bonus is some value given in case that $g()$ is a static function. The MI is optimistic it assumes most functions are not referenced externally and that additional indirect calls to a function can be detected and hence inlined-avoided. Moreover, if there are external calls, the user could copy these functions to a common header file so that they will be inlined in the other module. The MI, therefore, assumes that if all calls to a function have been inlined then the body of this function will not remain in the executable. The MI, therefore, inline a CS ($f \rightarrow g$):

- In case that there is only one call to g that is not in a conflict with a more profitable neighboring node, the MI will always inline; in comparison, the Oz may refrain from inlining if the body of g is too big. A simple program with a chain of calls (all nodes are tree nodes) was used to demonstrate the difference. Figure 6 depicts the call graph G of a given program. Here the saving on the edges contains two numbers: the left number is the cost in IRs as computed by the Oz inliner, and the right number is the saving in bytes as computed by the MI. The Oz will inline only if the cost (left number) is less equal to the threshold (5). None of the edges in figure 6 is profitable for Oz; consequently, the Oz did not inline any of the callees in the chain. The MI will inline all the tree-nodes as none of them is conflicting (the star at $f()$ is not profitable) and obtained the inlined program of figure 7. Consequently, compared to the Oz, the MI reduced the code size from 1266 bytes to 1154 bytes.



Figure 7: The Same program after MI's inlinings.

Figure 6: Program with a chain of calls

- In case that there is $k > 1$ calls to the same function (a star), the MI will either inline all the k calls or none depending if the total saving is greater than $(k - 1) \cdot |callee|$ or not. The Oz examines each of the star's calls separately and decides to inline if the coast (the left number on edge) is less equal the $threshold = 5$. This can lead to two problems:
 - The Oz will refrain from inlining the star as none of its call-site is profitable despite

the fact that inlining the whole star is profitable. This is depicted in figure 8 showing a star at $f1()$ with two calls from $main()$. As the Oz cost for each call is 10, the Oz will inline none of $f1()$ calls. However, for the MI the $f1$ -star is profitable, and compare to the Oz, the MI inlined the star reducing the size from 1036 bytes to 1020.

- The Oz will inline the stars calls since each one is profitable; however, this doesn't seem right since the start as a whole is not profitable. This is depicted in figure 9 showing a star at $f1()$ with ten calls from $main()$. All these calls are inlined by the Oz while the MI avoids inlining them ($9 \cdot 10 \leq 10 \cdot 9$). Reducing the size from 1324 bytes to 1172.

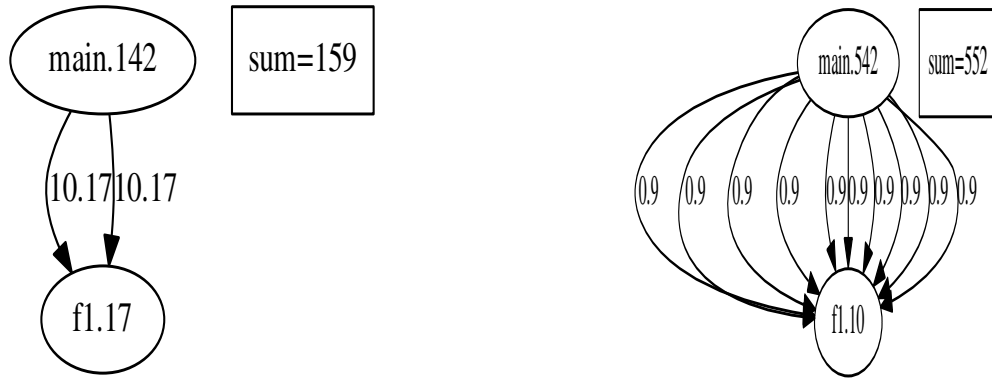


Figure 8: Oz refrain from inlining a star which is in fact profitable

Figure 9: Oz inline a star which is in fact profitable.

The mutual effect and multiple-rounds- Inlining one CS may render the usefulness of inlining another neighboring CS, which potentially could be more profitable than the first one. The MI checks such cases and selects the most profitable CS out of a conflicting set of CSS while the Oz checks each CS separately. In addition, the MI performs repeated rounds of inlining over G until no profitable rounds are found, while the Oz performs only one round of inlinings. Note that after each inline, the MI updates G , recomputing the callee body size and saving the neighboring CSS. Figure 10 depicts a program whose inline requires checking the mutual inlining effect and performing multiple inlining rounds. Here the Oz inline $f1() \rightarrow g1()$ following the bottom-up inline order. This inline prevents the inlining of $g1() \rightarrow q1()$ as $g1 + f1()$ is now too big, and indeed, the Oz does not perform any more inlinings. The MI's mutual inline effect prevent it from inlining $f1() \rightarrow g1()$ first and instead the MI first performs the inline of $q1() \rightarrow main()$ resulting by pushing the star at $g1()$ into main, i.e., $main + q1()$ now contains the four calls to $g1()$. This passes constant parameters to $g1()$ calls

making its inlining profitable. Thus next inline-round of MI will inlining all the remaining calls reducing the size from 1124 bytes to 1052.

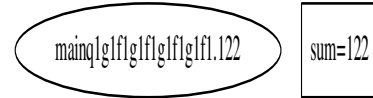
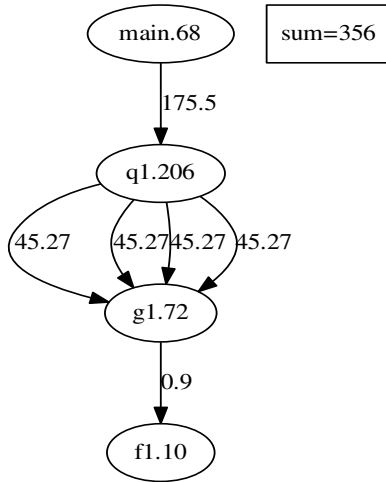


Figure 11: Final outcome.

Figure 10: A program demonstrating mutual effect and multiple rounds

5 Results

Results are given in the following table for a set of selected programs some from SPEC and some where arbitrarily selected from what was available at hand. The executable size where obtained using RISC-V-unknown-elf-size measures the size in bytes adding the *text* + *data* + *bss* sizes. The Oz row indicates the sizes obtained by the LLVM's inliner with -Oz option used for minimizing the executable size. While the MI row shows the size obtained by the MI algorithm. Some of the programs are in pure C-language and some are in C++ (as the inliner is part of the LLVM's optimization and hence works for any language supported by the LLVM).

Desc Calculator

	text	data	bss	total	diff
Oz	75928	2504	124	78556	
MI	75528	2504	124	78156	400

SPEC BZIP2:

Oz	86066	5592	4380	96038	
MI	85710	5592	4380	95682	356

Compiler GCC :

Oz	2906454	7428	561100	3474982	
----	---------	------	--------	---------	--

Particle simulation NBODY :

Oz	18581	0	40	18621	
MI	18553	0	40	18593	28

SPEC MCF

Oz	38536	2484	6244	47264	
MI	38356	2484	6244	47084	180

Highlevel synthesis tool ONE

Oz	65552	2486	96456	164494	
MI	65212		2486	96456	164154 340

SPEC HMMER

Oz	162400	4874	81604	248878	
MI	161988	4874	81628	248490	388

SPEC Soplex

Oz	572922	127331	6460	706713	
MI	577354	123397	6460	707211	-498

6 Related works

the code bloat constraints can be mapped to the knapsack problem, which has been shown to be NP-complete [5]. [6] use inlining trails wherein the expected benefit of a possible inline can be computed by evaluating the costs and benefits resulting by pseudo-inlining a given call site. The saving in code size is determined by “group analysis”, i.e., accounting for the effect of passing known static values into the callee’s parameters (as explained, this is also used in LLVM and consequently in the method described in this work). [7] studies aggressive inlining and how it affects the performance and shows that such an inline improves performance significantly. [8] study different strategies for inlining different versions of the callee, where a version refers to how many inlines (from previous steps) this callee contains. It proposes to use a greedy strategy wherein, at each step, they select the call site that saves the maximal number of dynamic calls. [9] propose a branch-and-bound search algorithm to find a subset of n given functions such that the resulting code size is less than a given limit and that the performance is maximized. They assume that performance is related to the number of calls that occurs during execution; hence the selected subset of functions to be inlined should minimize this number. Profiling and simulation are used to determine the expected performance. [10] studies the effect of static and profile-based inlining heuristics for the Jalapeño dynamic optimizing compiler for Java. They also consider the problem of finding the most performance-profitable subset of inlinings under a restricted code-size budget and formalize this inlining optimization problem as a variant of the Knapsack problem. This work is mainly concerned with building the call graph and obtaining profile information to handle dynamic class-loading in Java programs. [11] improves the “temperature” inlining heuristic of the ORC compiler, where “temperature” combines the time spent in a procedure and the size of the procedure. They modify the temperature such that more inlinings will be performed more aggressively for small-size

benchmarks than large-size benchmarks and decrease the inlining of callees containing loops with a high trip count.

[12] uses a fast classifier called random forests to optimize inlining performance the execution time is a module that measures the longest execution path in the control flow graph of the compiler (called WECT analysis).

[13] compares three inlining techniques genetic algorithm, trained neural networks and using a pruned decision tree. All three methods are based on measuring the values of an elaborate set of features such as: caller/callee memory operations, execution frequency, caller/callee nested loops, and methods invocation. Their results (speedup gain) on specJVM 2008 show that all three methods had the same impact.

References

- [1] Manish Verma, Lars Wehmeyer, and Peter Marwedel. Dynamic overlay of scratchpad memory for energy minimization. In *Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 104–109, 2004.
- [2] Wikipedia contributors. Inline function, 2019.
- [3] Chris Lattner and Vikram Adve. Llvm: A compilation framework for lifelong program analysis & transformation. In *International Symposium on Code Generation and Optimization, 2004. CGO 2004.*, pages 75–86. IEEE, 2004.
- [4] Gen-Huey Chen, MT Kuo, and JP Sheu. An optimal time algorithm for finding a maximum weight independent set in a tree. *BIT Numerical Mathematics*, 28(2):353–356, 1988.
- [5] Robert W Scheifler. An analysis of inline substitution for a structured programming language. *Communications of the ACM*, 20(9):647–654, 1977.
- [6] Jeffrey Dean and Craig Chambers. Towards better inlining decisions using inlining trials. In *Proceedings of the 1994 ACM Conference on LISP and Functional Programming*, pages 273–282, 1994.
- [7] Andrew Ayers, Richard Schooler, and Robert Gottlieb. Aggressive inlining. *ACM SIGPLAN Notices*, 32(5):134–145, 1997.
- [8] Owen Kaser and CR Ramakrishnan. Evaluating inlining techniques. *Computer Languages*, 24(2):55–72, 1998.
- [9] Rainer Leupers and Peter Marwedel. Function inlining under code size constraints for embedded processors. In *1999 IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers (Cat. No. 99CH37051)*, pages 253–256. IEEE, 1999.
- [10] Matthew Arnold, Stephen Fink, Vivek Sarkar, and Peter F Sweeney. A comparative study of static and profile-based heuristics for inlining. In *Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization*, pages 52–64, 2000.
- [11] Peng Zhao and José Nelson Amaral. To inline or not to inline? enhanced inlining decisions. In *International Workshop on Languages and Compilers for Parallel Computing*, pages 405–419. Springer, 2003.

- [12] Paul Lokuciejewski, Fatih Gedikli, Peter Marwedel, and Katharina Morik. Automatic wcet reduction by machine learning based heuristics for function inlining. In *3rd workshop on statistical and machine learning approaches to architectures and compilation (SMART)*, pages 1–15, 2009.
- [13] Sameer Kulkarni, John Cavazos, Christian Wimmer, and Douglas Simon. Automatic construction of inlining heuristics using machine learning. In *Proceedings of the 2013 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pages 1–12. IEEE, 2013.

VIRTUALISED ECOSYSTEM TO ENVISAGE NUMEROUS APPLICATIONS ON AN AUTOMOTIVE MICROCONTROLLER

Meghashyam Ashwathnarayan, Vaishnavi J,
Ananth Kamath and Jayakrishna Guddeti

Infineon Technologies India Pvt Ltd, 11MG Road, Bengaluru, Karnataka, India

ABSTRACT

In automotive electronics, new technologies are getting integrated into basic framework creating ways for new software-defined architectures. Virtualization is one of most discussed technologies which will offer a functionality growth into architecture of automobile. This paper introduces concept of validating test cases from multiple IPs on virtualised framework and also investigate the feasibility of implementing protection mechanism on memory segment dedicated for a virtual machine (VM). We describe a proof-of-concept which can be used to promote the use of virtualisation to extend the coverage of post silicon validation. Experimental results are presented as a quantitative evaluation of using virtualization for different testcase scenarios.

KEYWORDS

Virtualisation, Automotive, Multi-Core Systems, Hypervisor, Post Silicon Validation.

1. INTRODUCTION

Automotive industry at present is seeing a tremendous change. The latest connected technology trends are forcing old, purpose-built embedded systems to be modified or replaced by new technologies that defines a new dimension for software-based architecture. As functionality of automobile increases, the number of functional hardware modules and software used by automakers grows [1] [2]. Handling the rise in number of ECUs and complexity in an automobile has become a challenge for manufacturers (OEMs) [3]. Recently, there have been a lot of discussions about trends that can offer reduction of ECUs and to develop safe modules without restricting costumer's requirement [4]. Implementing new architectures into old which can help us manage complexity, power consumption, cost and weight. One emerging architecture in this industry is virtualisation. In [6][4], Gernot Heiser briefly talks about automotive industry with virtualisation in near future and benefits of this approach.

Virtualisation, is a technology by which multiple virtual machines (VMs) are multiplexed on single hardware machine ensuing a logical division of available physical resources. Virtualisation makes it possible to assign the hardware resources to multiple isolated applications. This is an effective approach to restructure the existing architecture, take full benefit of the performance of processors and concentrate on the increasing complexity of software-defined functions in vehicle. This project aims to implement framework for development and validation of testcases using concept of virtualization. In testcases requiring the fulfilment of safety standards, it is important to include the related safety aspects in the workflow which is offered by the addition of protection mechanisms. Since high-quality testcases for post-silicon validation should be prepared before a silicon is available in order to reduce time spent on preparing the tests,

debugging and fixing it after the silicon is available. We propose an approach of executing of post-silicon validation tests on virtual machines for improved test coverage. The proposed workflow should be able to provide parallel validation of multiple IP modules, while not decreasing the efficiency of the workflow. We also intend to understand potential savings in time and bill of materials used during validating both framework as well as IP modules.

2. WHAT IS VIRTUALISATION?

Virtualisation creates a virtual environment that replicates functionality to that of physical hardware machine. A virtual machine (VM) is an environment that is mounted on software, which synchronizes the execution of the VM's dedicated hardware. A software layer that resides between the VMs and the hardware is known as Hypervisor or Virtual Machine Manager (VMM). In [7], Popek and Goldberg highlights the following essential characteristics of a VMM:

- [1] Program running under the VMM should display expected functioning that is identical to that of demonstrated behaviour when running directly on the underlying hardware platform.
- [2] Overwhelming load should not degrade performance of VMM.
- [3] VMM should have complete control of the physical hardware resources which are allocated to the guest OS at all times.

Virtualised framework typically consists of a real-time component which performs critical tasks within a certain deadline and a general-purpose component that may contain processing information, configuring or managing the system.

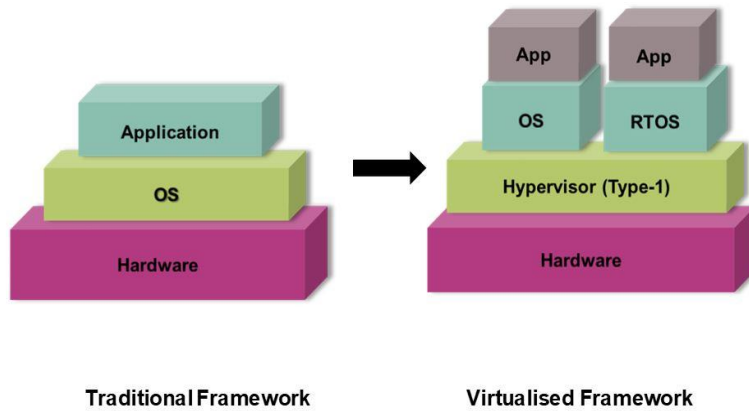


Fig. 1. Virtualised framework for Automotive domain

VMM software makes it possible to split the physical resources of a processor into safely separated segments for VMs mounted thus providing a near-complete isolation. Since VM is the logical replica of a physical hardware, multiple VMs can be installed on a same hardware and are logically separated providing isolation on the same hardware [5] [8]. The OS and application are completely unaware they are sharing hardware resources with other applications.

3. STATE OF THE ART

In the IT infrastructure, virtualisation is used to create multiple server instances from a physical server. Unlike a server that uses virtualisation to run many independent isolated servers on its VMs, embedded systems are highly integrated [9]. The prerequisites for a hypervisor for automotive systems is different from that of an IT domain. The requirements of hypervisor can be summarized as below [10] [9] [11].

VMM should incorporate processor architectures aiming embedded systems.

- [1] Support secure encapsulation of subsystem components that interact with another subsystem strongly.
- [2] Minimal impact on hardware resources and real-time performance such as fault-tolerance and time-related characteristics.
- [3] Supports a scheduling policy between VMs and provide priority for real-time system components.
- [4] Ensures strong spatial and temporal isolation between VMs by using concept of memory protection unit (MPU).

The prerequisite to integrate applications with very different requirements, modularly on a single processor that have the computing power to run these using a hypervisor [5].

4. VIRTUALISATION IN AUTOMOTIVE EMBEDDED SYSTEM

The total number of ECUs in the automotive network is growing due to the new functions which are implemented per ECU. Semiconductor vendors are also met with new challenges. Moore's Law defines that need for additional computing power and chip performance will double every 18–24 months. For the customer's comfort, demand for implementing cost-effective and innovative function in cars are growing. Over years of discussion on whether running multiple function on an ECU is a solution to reduce the ever-increasing digits of computing devices inside an automotive vehicle has started to find certainty in future architecture [12]. Since auto-mobile is a high-complexity device, introducing this approach requires thorough consideration about requirements regarding safety, availability, security, performance and resource management. This discussion often leads to virtualisation associated with a hypervisor which provides freedom from interference of different domains in auto mobile. The understanding of incorporation of such a technology on a single core is already complex. A challenge that we face is to integrate automotive customized functions into this new technology. Since Virtualisation presents the opportunity for multicore complexity and layer of abstraction for hardware, thus it offers multiple opportunities automotive industry [13].

A. Virtualised Framework

The framework envisages a virtualised isolated framework containing four VMs. In one of the derivatives of TC4x microcontroller, three hardware resource partitions (HRP) are utilised, the first HRP is used by the Hypervisor, the second HRP is used by the real-time VM1 and the third HRP is used by non-real-time VMs (VM2 to VMx). The figure virtualised ecosystem depicts a multicore System on Chip (SoC), a VMM and four VMs assigned to Core0. Each subsystem in a VM have four IPs being used to create an application. The IPs can be reused in different VMs, however a strict isolation of memories, sharing of resources should be enforced. In order to test the above concept of running multiple subsystem on a virtualised framework, a demo setup was created. Majority of the subsystems in independent VMs use Direct Memory Access (DMA) as

one of the IP used in the application. The DMA IP, as it is shared by all the VMs need to be partitioned in a way that the no channel is overlapped by another VM. A set of channels are assigned to a particular VM and are thoroughly isolated from each other using access protection offered in A3G.

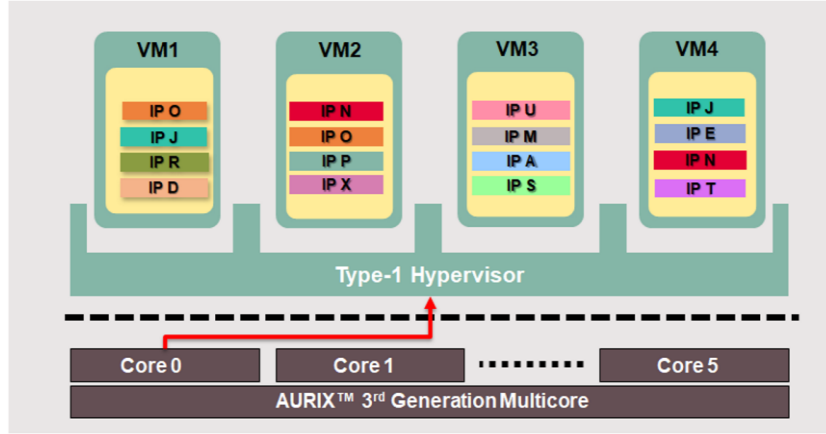


Fig. 2. Virtualised Ecosystem

5. VALIDATION IN VIRTUALISED FRAMEWORK

Post Silicon Validation commonly involves developing test cases for validating a design, analysing the results, involving in debug with assistance of designers. A significant amount of study is focused on detecting bugs in these silicon chips. With our approach of using virtualisation framework, it employs us to accelerate the validation with ease as various testcases can be deployed into multiple VMs and with no interference between the VMs, each testcase works independently. Every VM that is enabled will have application to be tested integrated on it. In figure 3 “Validation flow in AURIX™/TriCore™”, we can deploy four different testcases of different IPs on four different VMs. In our proposed framework, processor now has extended into four more VMs that copies processor functionality thus increased capability of software than before. With addition of protection mechanism for each VM, we deal with safety feature where it eliminates the possibility of the memories getting overlapped thus saving unnecessary traps while debugging. Near-complete isolation between application on the same hardware provides us advantage on testing multiple testcases. At present we are preparing the testcases by running on virtualised framework on emulator.

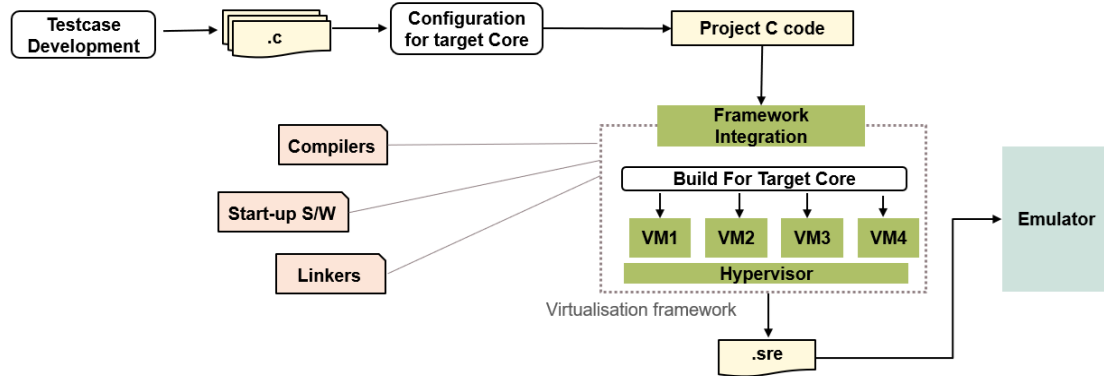


Fig. 3. Validation flow using virtualisation in AURIX™/TriCore™

We intend to run it on the chip once it is available. We have managed to find number of issues while integrating testcases into VMs. The main advantage that we have derived running these applications on an integrated platform was having a system level approach instead of having a directed test known to only validate a particular functionality/feature in an IP. Due to this, numerous issues lying in the interface of IPs have been found. The other positive is that the real-world applications can be deployed and validated in a virtualised environment there by increasing the confidence to the customer regarding virtualisation in the automotive field.

6. METHODOLOGY

Our main insight for this project is to enable application execution on a virtualised framework and to offer encapsulation to integrated applications to be tested and deployed in an automotive clustered environment. The main challenge when involving VMs lies in execution of an application with real-time constraints [5] and a framework that can support sharing of hardware resources between VMs. This requires a hypervisor with a scheduler which provides equal time slice for VMs enabled.

6.1. Hypervisor

AURIX™/TriCore™ uses Type-I VMM that is installed on the primary boot system of hardware and executes at the highest level of privilege with full control over any VMs that use it [14]. As Type 1 VMM often needs a scheduler to administer multiple VMs [15], a need for scheduling algorithm must be thoughtfully recommended in the hypervisor software where the priority is given to real time VM [16]. Generally, the VMM is executed in the privileged mode and hence has full access to hardware resources whereas VM is run under user processor mode. Multiple operating systems, serving the different needs of various subsystems, such as real-time environment and non-safety critical applications can be run on the virtualised framework [5]. With security sensitive aspect kept as a priority, we come up with best fit solution on how to use physical resources [8]. These set of channels service only that VM that it is assigned for. At no instance, the subsystems or application running in the VMs cause a conflict when requesting a DMA service.

When creating a subsystem, a section of memory range is assigned for the smooth operation of the VM. A high-level protection mechanism can also be setup such that when a non-assigned VM

tries to access a memory region which is out of its range an interrupt or an alarm is raised informing that it is an illegal access. These memory ranges are dedicated to a particular VM and is never shared by another VM hence is able to provide secure encapsulation for applications running on it. Another common IP that is predominantly used in the subsystems is the timer module, just like the DMA channel, each timer is setup to run independently by a VM without any conflict. Interrupt service routines (ISRs) are created independently for a VM. To avoid conflict of interrupt priority numbers, all the ISRs are assigned priority numbers mutually exclusive to one another.

6.2. Validation of DMA module in virtualised framework

Direct memory access (DMA) is a means of speeding up data transfer between peripheral device and processor's memory bus while reducing work overload on the CPU. The DMA implements virtualisation by extending the concept of HRP. In this approach, HRP are used to isolate sets of DMA channels from each other and provide isolation from interference of channels in other HRP when running functional operation. In TC4x, we have a number of resource partitions that can be assigned to any of the channels being used. Table 1 provides details on how channels of DMA0 is segregated into different resource partition along with its access master, virtual machine number, transfer operation and memory region in use.

Table 1. Resource partitioning of DMA in virtualised framework.

Ch No	VM No	HRP	Master	Src	Write Access (src)	Read Access (src)	Dst	Write Access (dst)	Read Access (dst)
0-15	VM1	RP0	DMA0	LMU0	No	Yes	LMU2	Yes	No
16-31	VM2	RP1	DMA0	LMU1	No	Yes	LMU3	Yes	No
32-47	VM3	RP2	DMA0	LMU4	No	Yes	LMU6	Yes	No
48-63	VM4	RP3	DMA0	LMU5	No	Yes	LMU7	Yes	No
64-79	VM1	RP4	DMA0	LMU0	No	Yes	LMU2	Yes	No
80-95	VM2	RP5	DMA0	LMU1	No	Yes	LMU3	Yes	No
96-111	VM3	RP6	DMA0	LMU4	No	Yes	LMU6	Yes	No
112-127	VM4	RP7	DMA0	LMU5	No	Yes	LMU7	Yes	No

6.3. Use of protection mechanism for DMA module

Each HRP has its own protection mechanism register which provides write protection to the HRP access registers (AR), whereas AR provides read and write protection to HRP specific registers which controls data transfer operation of DMA. Whether or not a channel is allowed to write or read to or from memory range is allocated in the access registers. To prevent unexpected use of memory, the access register memory range can be modified to restrict accesses to a certain range per channel/HRP. The memory region in which the HRP is configured can generate alarms or trap and deny any illegal access from other channel, which lies in a different HRP.

Function of access protection in DMA in a nutshell:

- 1) Attempt to access for DMA by unauthorized master leads to a trap.
- 2) It allows transfer operations which obey the access policy.
- 3) DMA channels under HRP configured by protection mechanism can request for DMA service.

6.4. Sequence of execution

- 1) Start-up software and firmware gets executed on the microcontroller. Necessary configurations with respect to virtualisation are configured.
- 2) Once control is in the VMM, it initialises the states and allocates the resources of the VMs which are enabled.
- 3) The VMM starts scheduling VM1 for execution.
- 4) DMA0 channels which are assigned in HRP0 and HRP4 are executed to transfer data from respective source and destination addresses.
- 5) After the DMA0 channels have finished their respective transactions, PPUC is scheduled as service by VM1. PPUC executes a complex fast Fourier transform (FFT) algorithm. Once the service is relinquished the control goes back to VM1.
- 6) A VMM call is made from VM1 so that the control gets to VMM and can schedule the next VM i.e. VM2.
- 7) This sequence of scheduling tasks by VMs continue until all the enabled VM's have performed their operations.

7. ADVANTAGE OF VIRTUALISATION IN AUTOMOTIVE DOMAIN

Introducing virtualisation into framework enables advantage [17] for vehicular architectures such as:

- A. **Hardware isolation:** Virtualisation provides the reusability of software. It also enables the upgrade of outdated infrastructure or application. It improves the portability of application or function to different hardware and OS platforms.
- B. **Hardware costs:** Virtualisation reduces hardware costs by enabling consolidation into a shared hardware resource. Efficiency of the hardware is improved as processor time, memories as well as peripherals are being shared by application running on different virtual machine. It allows users to take full benefit of concurrency provided by a multicore architecture.
- C. **System management:** Merging multiple ECUs will reduce hardware required which in turn decreases the production and operating costs. As multiple ECUs are combined together the power consumption in the vehicle is significantly reduced.
- D. **Isolation:** As mentioned in [7], VMM should be able to provide isolation of VMs. Isolation is also provided to each VM to operate independently without causing an impact to the system whenever any of the VM fails. This can be achieved by using timers in VMM such that if certain VM is unable to complete its function at assigned execution time, interrupt can be generated to request VMM to take necessary actions. When an application running on a VM faces failure, this failure does not propagate to the system and interfere during execution of other VMs.
- E. **Reliability and robustness:** The isolation and modularity provided by VMs improve reliability and robustness by reducing the effect of failure on a single VM.
- F. **Safety and security:** With implementing appropriate protection access mechanisms, it is able to offer secure encapsulation where interference between subsystems is reduced.

8. RESULT

The results in this section is regarding the correctness and performance evaluation of the DMA functionality running on framework. Firstly, for framework validation, we have provided snapshot from emulator that illustrates basic print statements in VMs of core0 which then was customized as per fig 3. With this, we are trying to validate whether VMs enabled by user are getting switched once finished its application without any error.

Secondly, we added DMA testcases on to the VMs to evaluate on how framework works. The main objective of running DMA module would be ease of validating data packet transaction with function that checks the data stored in both src and dst addresses using protection mechanism for DMA accesses as mentioned on table 1.

Finally, to get the proof of the DMA transferring operation, status register of DMA has been checked along to verify data is sent accordingly for respective LMU addresses. The desired results were obtained and we also saw generation of LMUaccess error when channels are accessed out of set memory region.

```
[dut_tb.inst_performance_counter_dut:3] started
53595 | 911.106 uS | TriCore CPU0.1 | % TC0 VM1
54051 | 915.666 uS | TriCore CPU0.2 | % TC0 VM2
56345 | 938.606 uS | TriCore CPU0.3 | % TC0 VM3
58985 | 965.006 uS | TriCore CPU0.4 | % TC0 VM4
61481 | 989.966 uS | TriCore CPU0.5 | % TC0 VM5
64023 | 1015.386 uS | TriCore CPU0.6 | % TC0 VM6
66491 | 1040.066 uS | TriCore CPU0.7 | % TC0 VM7

V C S   S i m u l a t i o n   R e p o r t
Time: 30000000000 ps
CPU Time: 317.490 seconds;      Data structure size: 6.7Mb
```

Fig. 4. Emulator snap of 7 VM shifting

9. CONCLUSIONS AND FUTURE SCOPE

Virtualisation approach tends to be efficient, which provides a benefit to use minimalistic hardware furthermore improving its security and CPU utilization. With a brief introduction to virtualisation techniques, we have presented a discussion on how our framework proposes testcases to run on VMs with additional protection mechanism to avoid illegal access. In our future research, we will explore on how to use this framework to its fullest which might include execution of automotive IP application such as non-safety critical applications such as fuel gauge monitor, battery level indicator, tyre pressure indicator and temperature monitor in-place of testcases while reducing standard footmark of IPs being used furthermore reducing the bill of materials while creating these subsystems.

10. ACKNOWLEDGMENTS

The authors thank Mr. Shyam Kommajosyula, head of the post-silicon validation department, Infineon Technologies India, Bengaluru and Pankaj Moharikar, line manager for their support in publishing this paper.

Acronym List

Acronym	Definition
AR	Access Registers
APU	Access Protection Unit
A3G	AURIX™/Tricore™ 3rd Generation
CPU	Central Processing Unit
DMA	Direct Memory Access
ECU	Electronic Control Unit
FFT	Fast Fourier Transform
HRP	Hardware Resource Partition
IP	Intellectual Property
ISR	Interrupt Service Routine
IT	Information technology
LMU	Local memory unit
MPU	Memory Protection Unit
OEM	Original Equipment Manufacturer
OS	Operating system
PPUC	Parallel processing unit controller
PROT	Protection Register

REFERENCES

- [1] M. Broy, "Challenges in automotive software engineering," in Proceedings of the 28th international conference on Software engineering, 2006, pp. 33–42.
- [2] H. Hanselmann, "Challenges in automotive software engineering," in Companion of the 30th international conference on Software engineering, 2008, pp. 888–888.
- [3] Y. Onuma, Y. Terashima, and R. Kiyohara, "Ecu software updating in future vehicle networks," in 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2017, pp. 35–40.
- [4] D. Reinhardt, D. Kaule, and M. Kucera, "Achieving a scalable e/e architecture using autosar and virtualization," SAE International Journal of Passenger Cars-Electronic and Electrical Systems, vol. 6, no. 2013- 01-1399, pp. 489–497, 2013.
- [5] N. Navet, B. Delord, M. Baumeister et al., "Virtualization in automotive embedded systems: an outlook," in Seminar at RTS Embedded Systems. Citeseer, 2010.
- [6] G. Heiser, "The role of virtualization in embedded systems," in Proceedings of the 1st workshop on Isolation and integration in embedded systems, 2008, pp. 11–16.
- [7] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," Communications of the ACM, vol. 17, no. 7, pp. 412–421, 1974.
- [8] M. Strobl, M. Kucera, A. Foeldi, T. Waas, N. Balbierer, and C. Hilbert, "Towards automotive virtualization," in 2013 International Conference on Applied Electronics. IEEE, 2013, pp. 1–6.
- [9] R. Mijat and A. Nightingale, "Virtualization is coming to a platform near you," ARM white paper, vol. 20, 2011.
- [10] I. Pavic and H. Dzapo, "Virtualization in multicore real-time embedded systems for improvement of interrupt latency," 05 2018, pp. 1405–1410.
- [11] M. Traub, A. Maier, and K. L. Barbehon, "Future automotive architecture and the impact of it trends," IEEE Software, vol. 34, no. 3, pp. 27–32, 2017.
- [12] G. K. Thiruvathukal, K. Hinsin, K. Laufer, and J. Kaylor, "Virtualization for computational scientists," Computing in Science Engineering, vol. 12, no. 4, pp. 52–61, 2010.
- [13] J. E. Smith and R. Nair, "The architecture of virtual machines," Computer, vol. 38, no. 5, pp. 32–38, 2005.
- [14] D. Reinhardt and G. Morgan, "An embedded hypervisor for safety-relevant automotive e/e-systems," in Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014), 2014, pp. 189–198.

- [15] A. Aguiar and F. Hessel, "Embedded systems' virtualization: The next challenge?" in Proceedings of 2010 21st IEEE International Symposium on Rapid System Prototyping. IEEE, 2010, pp. 1–7.
- [16] Z. Gu and Q. Zhao, "A state-of-the-art survey on real-time issues in embedded systems virtualization," 2012.
- [17] J. Pelzl, M. Wolf, and T. Wollinger, "Virtualization technologies for cars," Tech. Rep., 2008.

AUTHORS

Meghashyam Ashwathnarayan Graduated from Electronics and Communication in 2006-07. Following this, worked at McAfee for 3 years before pursuing MS in VLSI at MIT, Manipal. Currently working in Infineon Technologies, Bengaluru. Areas of expertise is in the field of Automotive Microcontrollers Post-silicon Validation and Compute architecture. Has numerous papers/publications/journals.



Jayakrishna Guddeti Lead Principal Engineer and design lead in microcontroller design and development across different functional areas, with total of 16 years of experience in Silicon Architecture, Digital Design and Post-Silicon Validation. Has 12 Patents and various IEEE publications.



Vaishnavi J received the M.E. degree in Embedded Systems from Manipal School of Information Science, Manipal in 2021. She currently works as a trainee engineer at Infineon Technologies, Bangalore. Her current interests include PCIe communication, virtualisation.



Ananth Kamath, Senior Specialist Software and Firmware Engineer at Infineon Technologies India. I am part of the Infrastructure Software team, my responsibilities includes low level drivers for AURIX™ microcontrollers, Application Start-up Software and reference demo applications.



DEEP LEARNING FRAMEWORKS EVALUATION FOR IMAGE CLASSIFICATION ON RESOURCE CONSTRAINED DEVICE

Mathieu Febvay and Ahmed Bounekkar

Université de Lyon, Lyon 2, ERIC UR 3083, F69676 Bron Cedex, France

ABSTRACT

Each new generation of smartphone gains capabilities that increase performance and power efficiency allowing us to use them for increasingly complex calculations such as Deep Learning. This paper implemented four Android deep learning inference frameworks (TFLite, MNN, NCNN and PyTorch) to evaluate the most recent generation of System On a Chip (SoC) Samsung Exynos 2100, Qualcomm Snapdragon 865+ and 865. Our work focused on image classification task using five state-of-the-art models. The 50 000 images of the ImageNet 2012 validation subset were inferred. Latency and accuracy with various scenarios like CPU, OpenCL, Vulkan with and without multi-threading were measured. Power efficiency and real-world use-case were evaluated from these results as we run the same experiment on the device's camera stream until they consumed 3% of their battery. Our results show that low-level software optimizations, image pre-processing algorithms, conversion process and cooling design have an impact on latency, accuracy and energy efficiency.

KEYWORDS

Deep Learning, On-device inference, Image classification, Mobile, Quantized Models.

1. INTRODUCTION

Nowadays, mobile devices are in every human hand, replacing slowly but surely our way of life. Many mobile applications use artificial intelligence in diverse ways such as gaming, social media, artistic filters or augmented reality using different tasks like face detection, real-time image classification or object detection. Unfortunately, many artificial intelligence models run in the cloud due to the computational resources needed to execute their complexity with millions of parameters. Today, more than ever, data privacy represents a major concern for people. On-device inference is an alternative, protecting data, fixing loss of internet connectivity, and reducing computing costs. However, computing power on these devices is clearly insufficient to run effectively and submitted to energy limitations.

Recent improvements made on hardware like Neural and Tensor Processing Unit (NPU/TPU), Digital Signal Processor (DSP), and other accelerators [1] let Machine Learning and Deep Learning on-device execution possible [2, 3]. Several mobile deep learning frameworks have been developed by open-source community or industry leader with low-level software optimization like General Matrix Multiplication (GeMM), GPU libraries (e.g. OpenCL™, Vulkan® and OpenGL® ES) and most recently general hardware accelerators API like NNAPI letting on-device inference become a new opportunity [4].

But these features are implemented differently in frameworks and combination of both model, framework, hardware and device make performance assessment difficult.

Two smartphones and one tablet, based on the two most popular architecture, Qualcomm Snapdragon and Samsung Exynos, were chosen. Android devices were selected because of its easier framework deployment process compared to Apple iPhone. We used four different frameworks with different low-level software optimization techniques such as integration of Arm assembly language code portion, integration of GeMM libraries Eigen, OpenBLAS or custom, NPU support and different software graphic libraries (OpenCL, OpenGL, Vulkan). Our models are pre-trained on ImageNet dataset with both Tensorflow and PyTorch allowing us to easily convert them to our two other frameworks.

Our approach is to evaluate frameworks and models designed and developed for mobile devices with the objective of providing the community our inference latency, Top-1 and Top-5 accuracy and power efficiency results of different models allowing scientists to take the proper decisions and save time when choosing software libraries and hardware in order to run image classification, object detection, instance segmentation on resource constrained devices based on Arm Cortex A architecture. Our work differs from other as we developed an Android Java application for each framework where inference took place.

2. RELATED WORK

Bahrampour *et al.* [5] evaluated Deep Learning frameworks performance but they focused their work on desktop computer with a Titan X GPU.

Lu *et al.* [6] launched their benchmark on different mobile frameworks with a Nvidia TK1 and TX1 which are not smartphones or tablet used by customers.

Sehgal and Kehtarnavaz [7] offered a benchmark of multiple deep learning models inferring on mobile SoC but they tested TFLite and Core ML only.

MLPerf [8] and AI Benchmark [9, 10] provide an Android application to test various models on the device using different scenarios. Limitations are the inference engine which is based on TFLite only and the output result, approximated (MLPerf) or displayed as a weighted score (AI Benchmark).

Bianco *et al.* [11] and Almeida *et al.* [12] proposed the most related works. They evaluated multiple models on diverse architecture among which there are mobile SoCs. The main difference is they didn't run their test from an Android application.

Benchmarks and previous work to evaluate the performance of deep learning models or frameworks on different devices exist but we propose an alternative approach as we focused our test on mobile devices, either smartphone or tablet, with frameworks and models optimized for them.

3. ALGORITHMIC APPROACH

For our experiment, we chose two smartphones which had a SoC generation gap and one tablet with a boosted SoC. Four frameworks were implemented on which we executed seven models, five 32-bit floating point and two quantized (8-bit integer) used as image segmentation backbones. To simulate the most representative use cases for real-time image segmentation tasks,

we needed a dataset with enough images. Our choice was to use the ImageNet 2012 validation dataset containing 50,000 images. We kept the results from this first benchmark to evaluate the device power consumption and the image inference latency from the device's camera.

3.1. Devices

We selected the latest Samsung Galaxy Tab S7 containing a Qualcomm Snapdragon 865+ SoC, the OnePlus 8 with a Qualcomm Snapdragon 865 and the newest Samsung Galaxy S21 with a Samsung Exynos 2100. The two Snapdragon are on the same architecture to explore if the extra 260 MHz on one big core and the 87 MHz boost on the GPU provided by the 865+ produce a significant impact on the latency. Recent release of the Exynos 2100 represents a generation gap with the Snapdragon 865. It's based on the new Arm Cortex X1 which, giving to Arm, is 30% faster and have twice the ML performance over the Cortex A77 [13]. The three devices have Android 11 operating system. Table 1 shows their specifications in-depth. Our experiment was launched on all hardware available on each device which was CPU, GPU and NPU/DSP with different hyper-threading scenarios. When we run on GPU, we inferred with OpenCL, OpenGL or Vulkan graphic libraries. Manufacturers consider CPU, GPU and NPU/DSP, as a whole, named the AI engine which can only run quantized models with specific software frameworks.

Table 1. Device SoC's specifications with quantity of RAM, type of cluster with number of cores in it, Arm reference and core frequencies

SoC	RAM (Gb)	Cluster	Number	Ref	Freq (GHz)
865	8	LITTLE	4	A55	1.80
		big	3	A77	2.42
		big	1	A77	2.84
865+	6	LITTLE	4	A55	1.80
		big	3	A77	2.42
		big	1	A77	3.10
2100	8	LITTLE	4	A55	2.20
		big	3	A78	2.80
		big	1	X1	2.90

3.2. Frameworks

We tested four open-source frameworks, TensorFlow Lite 2.4.0, MNN 1.1.0, NCNN 20201218 and PyTorch mobile 1.7.

They all had Arm NEON optimizations and OpenMP library integrated in their source code. TFLite [14] is, at the time of this paper, the only framework to have a general hardware accelerator library, NNAPI, which allow inference on the AI engine. MNN and NCNN use a custom GeMM implementation whereas PyTorch does not have a GPU and NPU inference option yet.

We selected these frameworks due to their mobile context. All of them are compatible with Android and iOS devices.

3.3. Models and Dataset

The inference was launched on ImageNet 2012 [15] pre-trained models commonly used as image segmentation backbone.

The main difficulty was to find different models available on both PyTorch and Tensorflow but we manage to download five 32-bits floating point models: SqueezeNet v1.1 (sqn11) [16], MobileNet v2 (mob2) [17], Inception v3 (inc3) [18], ResNet50 v1 (res50), ResNet101 v1 (res101) [19] and two TFLite quantized models: MobileNet v2 (mob2q) and Inception v3 (inc3q) to run on the AI engine.

Table 2 shows the Top-1 and Top-5 accuracy provided by Tensorflow and PyTorch Hub [20, 21, 22, 23].

Table 2. PyTorch and TensorFlow Top-1 and Top-5 model accuracies provided by the sources. Best accuracy for each model is in bold text

Framework	Model	Top-1 (%)	Top-5 (%)
PyTorch	SqueezeNet v1.1	58.19	80.62
	MobileNet v2	71.88	90.29
	Inception v3	77.45	93.56
	ResNet50 v1	76.15	92.87
	ResNet101 v1	77.37	93.56
Tensorflow	SqueezeNet v1.1	49.00	72.90
	MobileNet v2	71.90	91.00
	MobileNet v2 (quant)	70.80	89.9
	Inception v3	78.00	93.90
	Inception v3 (quant)	77.5	93.70
	ResNet50 v1	75.20	92.20
	ResNet101 v1	76.40	92.90

3.4. Model conversion process

The frameworks implemented for our experiment can't use the downloaded models, they need to be converted. TFLite and PyTorch mobile models were the easiest to switch because of the tools provided by their parent training framework but MNN and NCNN don't support all of the PyTorch and TensorFlow operations.

To be compatible, PyTorch models had to be converted in ONNX format. We run different converters to make them compatible with MNN and NCNN.

For Tensorflow models, the MNN and NCNN tools were unable to convert ResNet v1 and Inception v3 architecture.

3.5. Image pre-processing

During the training phase of our models, each image was transformed to fit in the input tensor. We had to reproduce the pre-processing steps to reproduce the best accuracy.

TensorFlow crops or pads the image to the littlest size followed by a scale down then it normalizes each image color channel, Red, Blue, Green, with mean and standard deviation equal to 127.5 for floating point models and mean to 0.0 and standard deviation to 1.0 for quantized models.

It is quite the opposite for PyTorch as it resizes the image before cropping or padding it. Its normalization parameters are respectively for red, blue and green channels and for mean: 0.485, 0.456, 0.406 and standard deviation: 0.229, 0.224, 0.225.

3.6. Algorithm

For each framework, we developed a Java Android application which looped on all the converted models and inferred each of the ImageNet 50,000 images for any hardware available (CPU, OpenCL, OpenGL, Vulkan or NNAPI) from one to ten threads.

At each inference the time elapsed by the device to output the probabilities was gathered. We compared the result to the image key contained in the ground truth file provided with the dataset to know if the highest probability and the five best were in it. Latency and accuracy are saved in a CSV file in the internal memory. When the test was launched, the device was plugged to the power source in plane mode and screen luminosity was at its minimum level. The energy consumption was not measure in this algorithm.

From the results collected in the previous algorithm, the same experiment parameters were executed from a camera stream acquired on the device. The energy efficiency of all the components as well as the image pre-processing time were evaluated. In addition, the screen and the camera power consumption were collected separately to isolate the hardware used during the inference.

Algorithm 1. Experiment algorithm

Input	image = 1,...,50000
Output	latency = inference latency of the image isInTop1 = ground truth compared to the best probability isInTop5 = ground truth compared to the five best probabilities
Parameters	hardware = CPU,...,NNAPI thread = 1,...,10 model = sqn11,...,inc3q

```

for hardware = CPU to NNAPI do
  for thread = 1 to 10 do
    for model = sqn11 to inc3q do
      for image = 1 to 50000 do
        preProcessedImage  $\leftarrow$  preProcessImage(image);
        startTime  $\leftarrow$  getSystemTime();
  
```

```

    probs ← infer(preProcessedImage);
    stopTime ← getSystemTime();
    latency ← (stopTime – startTime);
    descendantOrderSort(probs)
    isInTop1 ← false;
    isInTop5 ← false;
    if ground truth == probs[0] then
        | isInTop1 ← true;
        | isInTop5 ← true;
    end
    else if ground truth in probs[1:4] then
        | isInTop5 ← true;
    end
    appendToCSV(latency, isInTop1, isInTop5);
end
end
end
end

```

4. EXPERIMENTAL RESULTS

For our experiment, we chose two smartphones which had a SoC generation gap and one tablet with a boosted SoC. Four frameworks were implemented on which we executed seven models, five 32-bit floating point and two quantized (8-bit integer) used as image segmentation backbones.

4.1. ImageNet dataset latency

We ran Algorithm 1 on two smartphones and one tablet to get the closest real-world use case results. Our algorithm was looping on 50,000 images which could come closest to a video feed from the device camera to simulate an image segmentation backbone in real-time. An acceptable latency for this task is under 30 ms letting display around 30 frames per second while providing room for image pre-processing and decoding functions. One of the intrinsic limitations of our devices was the thermal protection mechanism also known as Dynamic Voltage Frequency Scaling (DVFS) or CPU throttling. The system downscales the CPU frequency to dissipate the heat. Figure 1 shows two different DVFS behaviours when we ran NCNN on the Snapdragon 865 with one CPU thread. DVFS effect of Inception v3 pre-trained with PyTorch (1a) is not obvious, resulting in a stable inference with a narrow range around 4 ms (1b). On the contrary, ResNet 50 v1 pre-trained with the TensorFlow framework (1c) shows two inference levels, 165 ms and 280 ms (1d). From the 30,000th image, the SoC is so hot it stands longer at 280 ms. DVFS is less

present on the Snapdragon 865+ because it is an 11-inch tablet which contains more space to dissipate the heat, unlike the two other devices as shown on Figure 2.

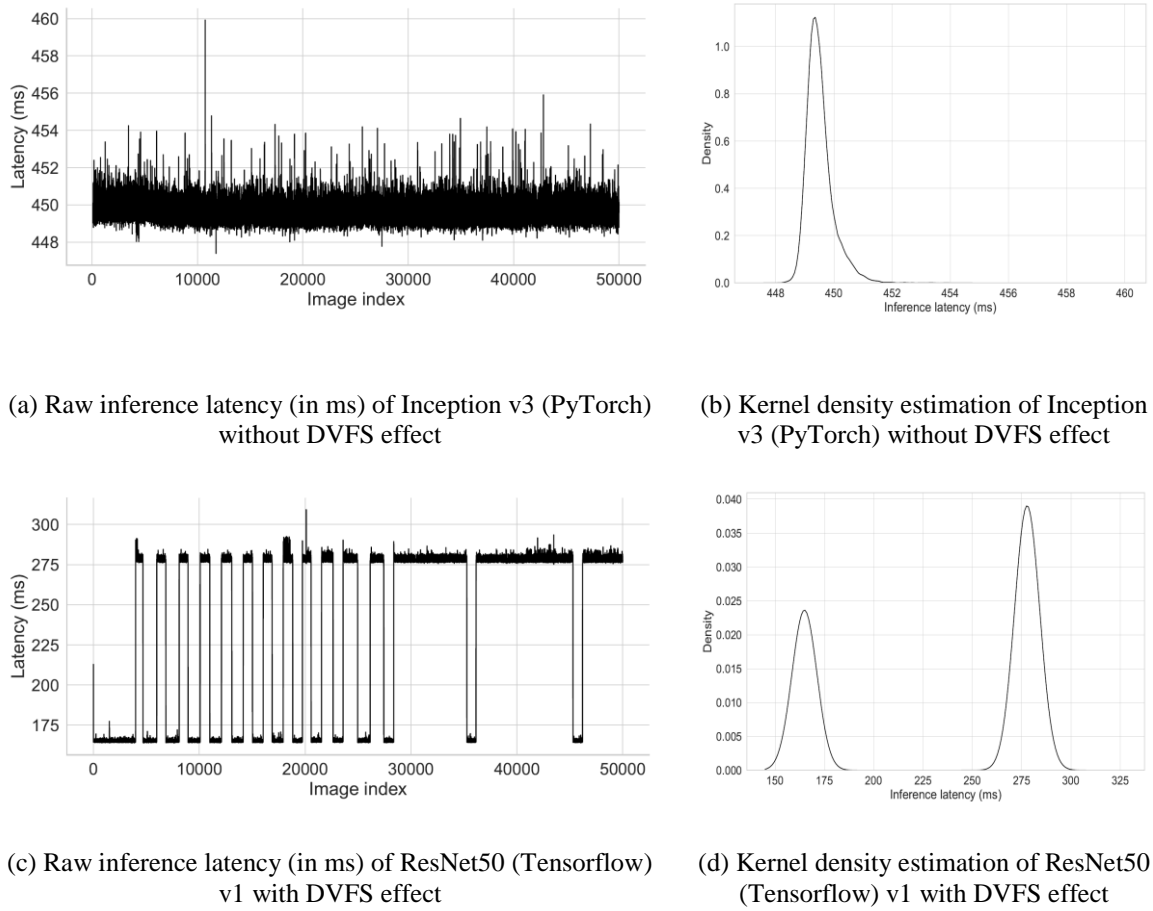


Figure 1. Inference without (a)(b) and with (c)(d) DVFS on Snapdragon 865 CPU with 1 thread

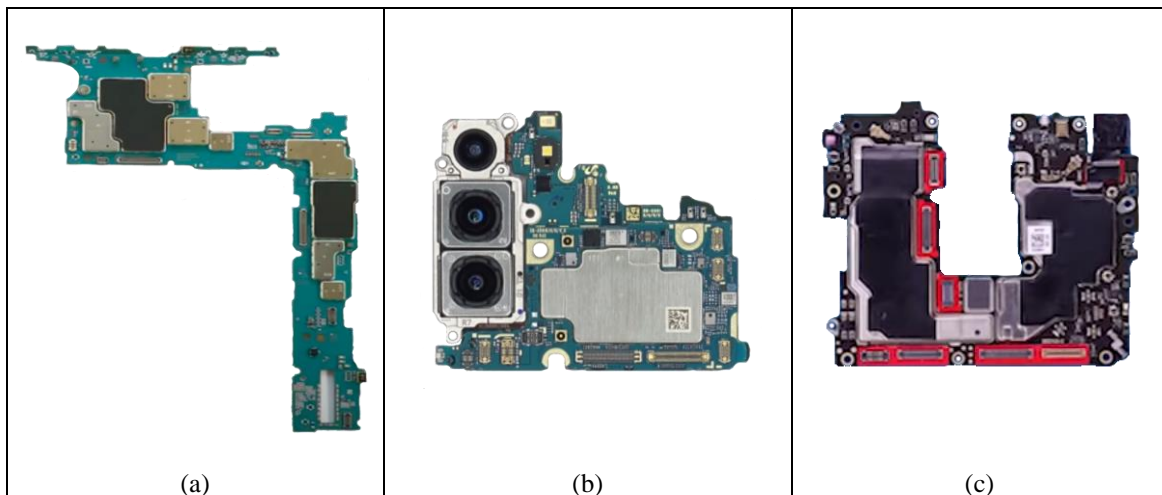


Figure 2. SoC's boards from Samsung Galaxy Tab S7 (a), Samsung Galaxy S21 5G (b) and OnePlus 8 (c) (not to scale)

Figure 3a shows that the multi-threading mechanism didn't affect the GPU. Switching from 1 to 10 threads didn't affect the latency.

Figure 3b shows the AI engine, which uses CPU, GPU and NPU.

We saw that MobileNet v2 and Inception v3 latencies were improved when switching from the GPU with floating point format to the AI engine with quantized one. Quantized version of Inception v3 on the Exynos 2100 is improved when running from 1 to 4 threads. The NNAPI library uses the best hardware in order to improve the latency. In our case, the library used the NPU, and the GPU excepted for Inception v3 model on the Exynos 2100.

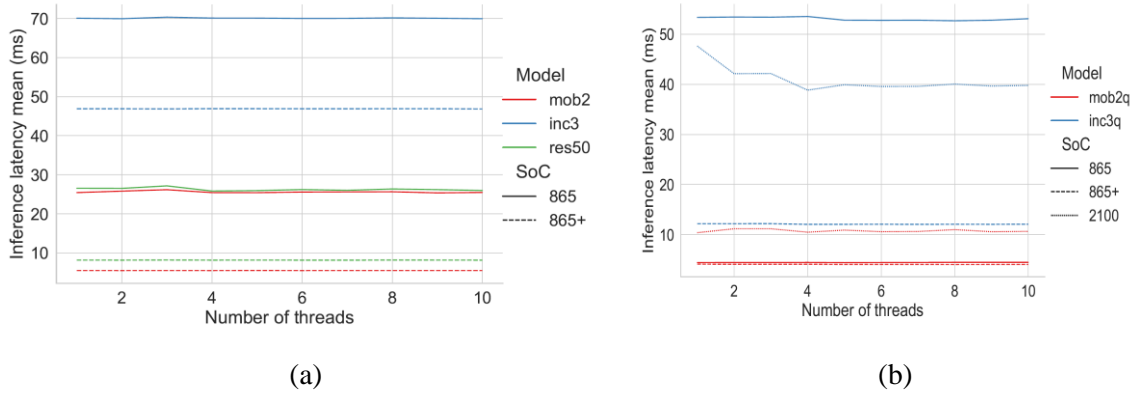


Figure 3. Influence of multi-threading on GPU (a) and AI engine (b)

Table 3 represents the arithmetic mean (μ) and the standard deviation (σ) of the inference latency in milliseconds with the accuracy loss compared to their reference model in Table 2. For each row we reported the best results of our experiment.

TensorFlow model latencies are the best with TFLite OpenCL for floating point models. We greyed SqueezeNet v1.1, ResNet50 v1 and ResNet101 v1 Tensorflow models in our table due to conversion issue reported in Section 3.4. The new Exynos 2100 provides an improvement in comparison of the Snapdragon 865 especially for PyTorch models inferred with NCNN Vulkan. NCNN has a non-negligible accuracy drop on different models but on SqueezeNet v1.1 it is 18 % faster than the second best framework, MNN, with 13.91 ± 0.16 ms on Snapdragon 865 and 13.36 ± 0.43 ms on Snapdragon 865+.

Snapdragon 865+ outperforms the most recent generation due to its better CPU and GPU frequency. This increase in frequency should represent a problem due to the throttling mechanism however the device demonstrates an excellent capability to dissipate the heat making extra computational power efficient.

The inconclusive results of the TFLite NNAPI on the 2100 should be related to the driver compatibility of the Samsung NPU which was probably unimplemented yet.

Table 3. Best mean (μ) with standard deviation (σ) of inference time in milliseconds for each model trained by TensorFlow (model-tf) and PyTorch (model-pt) of all hardware and framework on the three devices with their accuracy loss compared to Table 2 Top-1 and Top-5

SoC	Model	Framework	Hardware	$\mu \pm \sigma$ (ms)	Top-1 (%)	Top-5 (%)
Snapdragon 865	sqn11-pt	NCNN	CPU2	11.40 ± 3.89	-13.07	-10.67
	sqn11-tf	NCNN	CPU3	20.04 ± 4.13	-24.67	-28.25
	mob2-pt	MNN	CPU7	15.96 ± 2.32	-3.17	-1.49
	mob2-tf	NCNN	CPU3	13.00 ± 2.86	-3.79	-3.30
	mobq2-tf	TFLite	NNAPI	4.44 ± 0.69	-1.79	-1.15
	inc3-pt	MNN	CPU8	158.54 ± 33.94	-1.40	-0.68
	inc3-tf	TFLite	OpenCL	70.07 ± 1.69	-0.44	-0.24
	inc3q-tf	TFLite	NNAPI	52.67 ± 4.46	-0.26	-0.20
	res50-pt	NCNN	Vulkan	86.23 ± 2.87	-7.70	-4.22
	res50-tf	TFLite	OpenCL	26.54 ± 13.06	-47.08	-42.04
	res101-pt	NCNN	Vulkan	133.17 ± 2.24	-5.78	-3.09
	res101-tf	TFLite	OpenCL	25.98 ± 13.03	-48.28	-42.74
Snapdragon 865 +	sqn11-pt	NCNN	CPU3	10.95 ± 2.24	-13.07	-10.67
	sqn11-tf	TFLite	OpenCL	8.14 ± 0.59	-20.88	-22.74
	mob2-pt	NCNN	CPU3	14.54 ± 1.24	-9.51	-5.78
	mob2-tf	TFLite	OpenCL	5.47 ± 0.70	-1.70	-1.63
	mobq2-tf	TFLite	NNAPI	4.07 ± 0.81	-1.79	-1.15
	inc3-pt	MNN	CPU5	182.09 ± 23.17	-1.40	-0.68
	inc3-tf	TFLite	OpenCL	46.85 ± 0.60	-0.44	-0.24
	inc3q-tf	TFLite	NNAPI	12.06 ± 0.81	-0.26	-0.20
	res50-pt	NCNN	Vulkan	66.29 ± 2.75	-7.70	-4.22
	res50-tf	TFLite	OpenCL	8.13 ± 0.58	-47.08	-42.04
	res101-pt	NCNN	Vulkan	101.22 ± 2.3	-5.78	-3.09
	res101-tf	TFLite	OpenCL	8.17 ± 0.59	-48.28	-42.74
Exynos 2100	sqn11-pt	MNN	CPU4	12.88 ± 0.44	-4.17	-3.08
	sqn11-tf	TFLite	OpenCL	18.02 ± 7.46	-20.88	-22.74
	mob2-pt	MNN	CPU5	14.55 ± 3.43	-3.17	-1.49
	mob2-tf	TFLite	OpenCL	11.37 ± 6.80	-1.70	-1.63
	mobq2-tf	TFLite	NNAPI	10.77 ± 5.56	-1.79	-1.15
	inc3-pt	MNN	CPU4	194.73 ± 43.03	-1.40	-0.68
	inc3-tf	TFLite	OpenCL	93.05 ± 31.31	-0.44	-0.24
	inc3q-tf	TFLite	NNAPI	40.92 ± 9.53	-0.26	-0.20
	res50-pt	NCNN	Vulkan	81.48 ± 9.72	-7.70	-4.22
	res50-tf	TFLite	OpenCL	17.00 ± 6.71	-47.08	-42.04
	res101-pt	NCNN	Vulkan	117.25 ± 29.78	-5.78	-3.09
	res101-tf	TFLite	OpenCL	17.07 ± 7.01	-48.28	-42.74

4.2. Accuracy

An accuracy loss occurred when the model is converted. For Tensorflow models there was a drop of 1-2 % on Top-1 and 2-3 % on Top-5 on all frameworks with, from the lowest loss to highest: TFLite, MNN, NCNN. The same figures appeared for PyTorch ones except for NCNN which had a 7-13 % drop on Top-1 and a 5-11 % drop on Top-5 depending on models.

In addition, SqueezeNet v1.1, ResNet50 v1, ResNet101 v1 from TensorFlow were not operating the pre-processing parameters provided on TensorFlow Hub leading to an accuracy cap on both Top-1 and Top-5 with respectively 28.12 % and 50.16 % for them.

The accuracy loss from the quantized model is negligible regarding the latency gain. MobileNet v2 lost 1.89 % compared to its floating-point version but it reduced its latency by 5 % on the 2100 SoC, 26 % on the 865+ and 66 % on the 865. This gain is even bigger with Inception v3 model.

4.3. Camera stream latency

The camera stream from the device was integrated inside the Android application using Camera2 API. Images are acquired by the device camera in the YUV420 format and converted into ARGB8888 to make it compatible with models input. The image's size is 640 pixels width and 480 pixels height. These new outcomes integrate the image pre-processing latency executed by the framework and show CPU and GPU governors behaviour once the device is powered by its battery. These results are consistent with the ImageNet ones. There is a performance drop for all the frameworks as the device has to manage its energy. We observe that TFLite is more affected than MNN or NCNN. Once again, quantized models outperform the others on the three devices. It is particularly obvious for Inception v3 as it is approximately 3 times faster than its floating-point version on Snapdragon 865, 2.5 times on Snapdragon 865+ and 1.5 times on the Exynos 2100. This experiment confirms the performance of the Snapdragon 865+ related to a reduced DVFS effect.

4.4. Power efficiency

Before each test, devices were fully charged, screen brightness was set to medium, Bluetooth and Wi-Fi were turned ON to reproduce as much as possible real usage of the device. The test was stop once the device's battery reaches 97 % to avoid the nonlinear discharge of the lithium-ion battery.

We recorded the elapsed time for the device to go from 100 to 97 % with the help of Battery Historian software from Google [24]. We measured the screen consumption by setting the device in plane mode and recording the time for the device to reach 97 % when the screen is ON with medium brightness. Then we measured the camera consumption by doing the same process as for the screen but with launching the camera application. Then we subtracted the screen consumption to the observed one to have the camera.

The energy consumption for the Snapdragon 865, 865+ and Exynos 2100 screen are respectively 214 mAh, 619 mAh and 198 mAh. For the cameras, 438 mAh, 413 mAh and 792 mAh. The Snapdragon 865+ screen is bigger than the two others and the Exynos 2100 has the most powerful camera module.

Once again, our results show small models and quantized models are the more energy efficient. The faster it runs the less energy it consumes. Also, device screen and camera have a bigger impact on energy than the dedicated inference hardware.

Table 4. Latency and energy consumed in μA for processing one image from device camera stream after consuming 3 % of battery device. Hardware consumption is for the energy consumed by the hardware involved in the inference (CPU, GPU, NPU, RAM) and Device consumption represents the total of energy consumed by the device (screen and camera included).

SoC	Model	Framework	Hardware	$\mu \pm \sigma$ (ms)	Hardware consumption ($\mu\text{A}/\text{img}$)	Device consumption ($\mu\text{A}/\text{img}$)
Snapdragon 865	sqn11-pt	NCNN	CPU2	11.40 ± 3.89	2.73	6.55
	sqn11-tf	NCNN	CPU3	20.04 ± 4.13	5.69	9.76
	mob2-pt	MNN	CPU7	15.96 ± 2.32	0.64	4.53
	mob2-tf	NCNN	CPU3	13.00 ± 2.86	3.24	6.49
	mobq2-tf	TFLite	NNAPI	4.44 ± 0.69	0.92	3.69
	inc3-pt	MNN	CPU8	158.54 ± 33.94	27.40	59.78
	inc3-tf	TFLite	OpenCL	70.07 ± 1.69	17.06	34.22
	inc3q-tf	TFLite	NNAPI	52.67 ± 4.46	2.47	7.40
	res50-pt	NCNN	Vulkan	86.23 ± 2.87	13.11	31.55
	res50-tf	TFLite	OpenCL	26.54 ± 13.06	7.14	15.58
	res101-pt	NCNN	Vulkan	133.17 ± 2.24	20.06	48.12
	res101-tf	TFLite	OpenCL	25.98 ± 13.03	8.50	25.48
Snapdragon 865 +	sqn11-pt	NCNN	CPU3	10.95 ± 2.24	3.92	7.57
	sqn11-tf	TFLite	OpenCL	8.14 ± 0.59	2.97	8.06
	mob2-pt	NCNN	CPU3	14.54 ± 1.24	4.85	9.37
	mob2-tf	TFLite	OpenCL	5.47 ± 0.70	2.16	6.49
	mobq2-tf	TFLite	NNAPI	4.07 ± 0.81	0.94	5.09
	inc3-pt	MNN	CPU5	182.09 ± 23.17	51.73	107.44
	inc3-tf	TFLite	OpenCL	46.85 ± 0.60	11.44	30.98
	inc3q-tf	TFLite	NNAPI	12.06 ± 0.81	2.69	10.35
	res50-pt	NCNN	Vulkan	66.29 ± 2.75	18.88	39.22
	res50-tf	TFLite	OpenCL	8.13 ± 0.58	8.01	19.73
	res101-pt	NCNN	Vulkan	101.22 ± 2.3	34.17	70.97
	res101-tf	TFLite	OpenCL	8.17 ± 0.59	14.28	35.13
Exynos 2100	sqn11-pt	MNN	CPU4	12.88 ± 0.44	1.15	5.39
	sqn11-tf	TFLite	OpenCL	18.02 ± 7.46	1.89	13.18

mob2-pt	MNN	CPU5	14.55 ± 3.43	1.67	7.82
mob2-tf	TFLite	OpenCL	11.37 ± 6.80	3.21	14.99
mobq2-tf	TFLite	NNAPI	10.77 ± 5.56	2.96	13.70
inc3-pt	MNN	CPU4	194.73 ± 43.03	10.57	73.67
inc3-tf	TFLite	OpenCL	93.05 ± 31.31	21.73	60.38
inc3q-tf	TFLite	NNAPI	40.92 ± 9.53	4.30	30.48
res50-pt	NCNN	Vulkan	81.48 ± 9.72	14.17	39.63
res50-tf	TFLite	OpenCL	17.00 ± 6.71	7.18	33.19
res101-pt	NCNN	Vulkan	117.25 ± 29.78	15.63	54.00
res101-tf	TFLite	OpenCL	17.07 ± 7.01	15.18	52.90

5. CONCLUSION

In this paper we presented an inference latency benchmark on mobile to help the community better deployed image classification/segmentation model on Android devices. Our results showed that quantized models on AI engine should be the de facto standard, especially for complex models like Inception v3. Quantized models are more energy efficient and performs better than floating point ones with a tiny loss of accuracy. If there is no other choice than floating points, developers should go for TFLite. It experiences an easy model conversion and integration process on Android. For PyTorch models, we saw NCNN is a notable candidate, but it needs to improve its conversion process to gain more accuracy. We are looking forward to GPU and NPU/DSP's support in the future PyTorch mobile framework.

MNN and NCNN integration of these frameworks inside Android application is not a straightforward task. The conversion step is not user-friendly as engineer need to compile or find the appropriate converter and execute commands to transform the original model to a compatible and optimized one. Additionally, framework libraries must be compiled and integrated with the Android NDK which is an error prone process.

To conclude, manufacturers should improve heat dissipation or cooling mechanism on small devices to avoid the DVFS effect resulting in an improved latency.

REFERENCES

- [1] Albert Reuther, Peter Michaleas, Michael Jones, Vijay Gadepally, Siddharth Samsi, and Jeremy Kepner. Survey and benchmarking of machine learning accelerators. 2019 IEEE High Performance Extreme Computing Conference (HPEC), pages 1–9, 2019.
- [2] Sahar Voghoei, Navid Hashemi Tonekaboni, Jason G Wallace, and Hamid Reza Arabnia. Deep learning at the edge. 2018 International Conference on Computational Science and Computational Intelligence (CSCI), pages 895–901, 2018.
- [3] Carole-Jean Wu, David Brooks, Kevin Chen, Douglas Chen, Sy Choudhury, Marat Dukhan, Kim M. Hazelwood, Eldad Isaac, Yangqing Jia, Bill Jia, Tommer Leyvand, Hao Lu, Yang Lu, Lin Qiao, Brandon Reagen, Joe Spisak, Fei Sun, Andrew Tulloch, Peter Vajda, Xiaodong Wang, Yanghan Wang, Bram Wasti, Yiming Wu, Ran Xian, Sungjoo Yoo, and Peizhao Zhang. Machine learning at facebook: Understanding inference at the edge. 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), pages 331–344, 2019.

- [4] Mengwei Xu, Jiawei Liu, Yuanqiang Liu, Felix Xiaozhu Lin, Yunxin Liu, and Xuanzhe Liu. A first look at deep learning apps on smartphones. In WWW '19, 2019.
- [5] Soheil Bahrampour, Naveen Ramakrishnan, Lukas Schott, and Mohak Shah. Comparative study of deep learning software frameworks arXiv: Learning, 2016.
- [6] Zongqing Lu, Swati Rallapalli, Kevin S. Chan, and Thomas F. La Porta. Modeling the resource requirements of convolutional neural networks on mobile devices. Proceedings of the 25th ACM international conference on Multimedia, 2017.
- [7] Abhishek Sehgal and Nasser Kehtarnavaz. Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps. Machine Learning and Knowledge Extraction, 1:450–465, 2019.
- [8] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, Ramesh Chukka, Cody Coleman, Sam Davis, Pan Deng, Greg Diamos, Jared Duke, Dave Fick, J. Scott Gardner, Itay Hubara, Sachin Idgunji, Thomas B. Jablin, Jeff Jiao, Tom St. John, Pankaj Kanwar, David Lee, Jeffery Liao, Anton Lokhmotov, Francisco Massa, Peng Meng, Paulius Micikevicius, Colin Osborne, Gennady Pekhimenko, Arun Tejusve Raghunath Rajan, Dilip Sequeira, Ashish Sirasao, Fei Sun, Hanlin Tang, Michael Thomson, Frank Wei, Ephrem Wu, Lingjie Xu, Koichi Yamada, Bing Yu, George Yuan, Aaron Zhong, Peizhao Zhang, and Yuchen Zhou. Mlperf inference benchmark, 2019.
- [9] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In ECCV Workshops, 2018.
- [10] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seung soo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pages 3617–3635, 2019.
- [11] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. IEEE Access, 6:64270–64277, 2018.
- [12] Mario Almeida, Stefanos Laskaridis, Ilias Leontiadis, Stylianos I. Venieris, and Nicholas D. Lane. Embench: Quantifying performance variations of deep neural networks across modern commodity devices. In The 3rd International Workshop on Deep Learning for Mobile Systems and Applications, EMDL'19, page 1–6, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367714. doi: 10.1145/3325413.3329793. <https://doi.org/10.1145/3325413.3329793>.
- [13] Arm Ltd. Introducing the arm cortex-x custom program, 2021. <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/arm-cortex-x-custom-program>.
- [14] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pages 265–283, 2016. <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>.
- [15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.
- [16] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. ArXiv, abs/1602.07360, 2016.
- [17] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. ArXiv, abs/1801.04381, 2018.
- [18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
- [20] Google. Tensorflow hub, 2020. <https://tfhub.dev/>.
- [21] Google. TFLite hosted models, 2020. <https://github.com/tensorflow/models/tree/master/research/slim>.
- [22] PyTorch. Pytorch model hub, 2020. <https://pytorch.org/hub/>.
- [23] TensorFlow. TFLite hosted models, 2020. https://www.tensorflow.org/lite/guide/hosted_models.
- [24] Google. Battery historian GitHub, 2021. <https://github.com/google/battery-historian>.

AUTHORS

Mathieu Febvay is currently a PhD candidate at ERIC laboratory at University of Lyon in France (UR 3083). His research focuses on Lightweight Deep Learning where he investigates performance and feasibility of neural networks model running on resource constrained devices. He also works as a Software Engineer on mobile devices. He holds a Master in Computer Science (MIAGE) from University of Lyon (2017) and is graduated in Software Development from University of Montpellier (2008). He has interest in the field of health and mobile medical devices.

Ahmed Bounekkar is an Associate Professor at the University of Lyon 1, attached to the ERIC laboratory. Since 2009, he has been in charge of the Master MIAGE in management informatics. His research focuses on modelling in complex systems for the design of decision support methodologies. They particularly concern the development of algorithms for data structuring, machine learning and multi-objective optimisation problems. The proposed models mainly concern problems in the field of health.

LEVERAGING OPENAMP IN EMBEDDED MIXED-SAFETY CRITICAL SYSTEMS

Mridula Prakash

L&T Technology Services, CTO Office, Mysore, India

ABSTRACT

This aim of this paper is to understand OpenAMP framework and provide details on how to use OpenAMP for designing mixed-safety critical systems. OpenAMP is an open-source software framework that provides software components for working with Asymmetric multiprocessing (AMP) systems. The paper explains the software components of OpenAMP and provides details to use OpenAMP for embedded system designs.

KEYWORDS

OpenAMP, Multicore, Mixed Critical, & Embedded Systems.

1. INTRODUCTION

The electronics industry has seen a dramatic rise in the use of multicore processing in recent years. Multicore designs in embedded systems are now becoming mainstream due to market demand for increased performance, lower power consumption, and optimized costs. CPUs with diverse capabilities are therefore being clustered together to optimally handle different tasks in a single System on Chip (SOC). Multicore systems will enable data and task parallelism, with each core working independently.

1.1. Multicore Systems

In homogenous computing, all core in a multicore CPU are identical and execute the same instruction set. To overcome limitation of homogenous computing it gave rise to heterogenous computing, where the cores are not identical and implement different instruction. Essentially, there are two multicore system architectures – Asymmetric Multicore Processing (AMP) and Symmetric Multicore Processing (SMP). SMP provides an approach to multicore design in which all cores share the same memory, operating systems, and other resources. AMP, on the other hand, is an approach to multicore design in which cores operate independently and perform dedicated tasks.

An AMP system may be constructed from any combination of core architectures; all the cores may be identical or there may be a rich mixture of core types that includes conventional processing units as well as specialized cores, like digital signal processors (DSPs) for instance. Each core executes independently in an AMP architecture, with or without an operating system, and their operating system may be selected based on the required functionality.

Combinations of SMP and AMP yield good results in scenarios in which the main system runs on a few cores that use SMP and are helped by cores running AMP modes as software accelerators. It should be noted here that multicore applications can be implemented using a SMP-enabled

operating system, but that approach does not allow for independent workloads to be executed on different cores and does not support utilization of heterogeneous cores.

1.2. Challenges with an AMP System

In an AMP system, there is no unified operating system or scheduler managing all resources in the system.

1. An inter-core communication facility issues that require the cores to be protected from one another.
2. Boot order – the sequence in which the software on each core starts – may be important to avoid synchronization and security issues.
3. Debugging the disparate workloads running on the potentially heterogeneous cores can be quite challenging.

Hence, there is a need for a software framework which helps manage boot order, data exchange, and power consumption.

2. OPENAMP

Open AMP came into existence to manage boot order and communication between the cores that are running independently.

OpenAMP stands for Open Asymmetric Multi-Processing, an open-source software framework that provides software components for working with Asymmetric multiprocessing (AMP) systems. The framework is maintained by the OpenAMP project, which comprises of member companies, including, Xilinx, ARM, STMicroelectronics, Linaro, Texas Instruments, Wind River and Nordic Semiconductor.

The key components and capabilities provided by the OpenAMP Framework include:

1. remoteproc – This component allows for the Life Cycle Management (LCM) of remote processors from software running on a master processor. The remoteproc API provided by the OpenAMP Framework complies with the remoteproc infrastructure that is present in upstream Linux 3.4.x kernel onward. It has the ability to load firmware and to start and stop remote processors.
2. RPMsg – The RPMsg API facilitates Inter Processor Communications (IPC) between the independent software contexts running on heterogeneous or homogeneous cores present in an AMP system. This API is compliant with the RPMsg bus infrastructure present in upstream Linux 3.4.x kernel onward.

The API enables applications to send and receive variable length binary message data, with the message format being defined by the application. It is well suited for exchanging asynchronous and event-based messages with remote processors.

2.1. OpenAMP Bootup

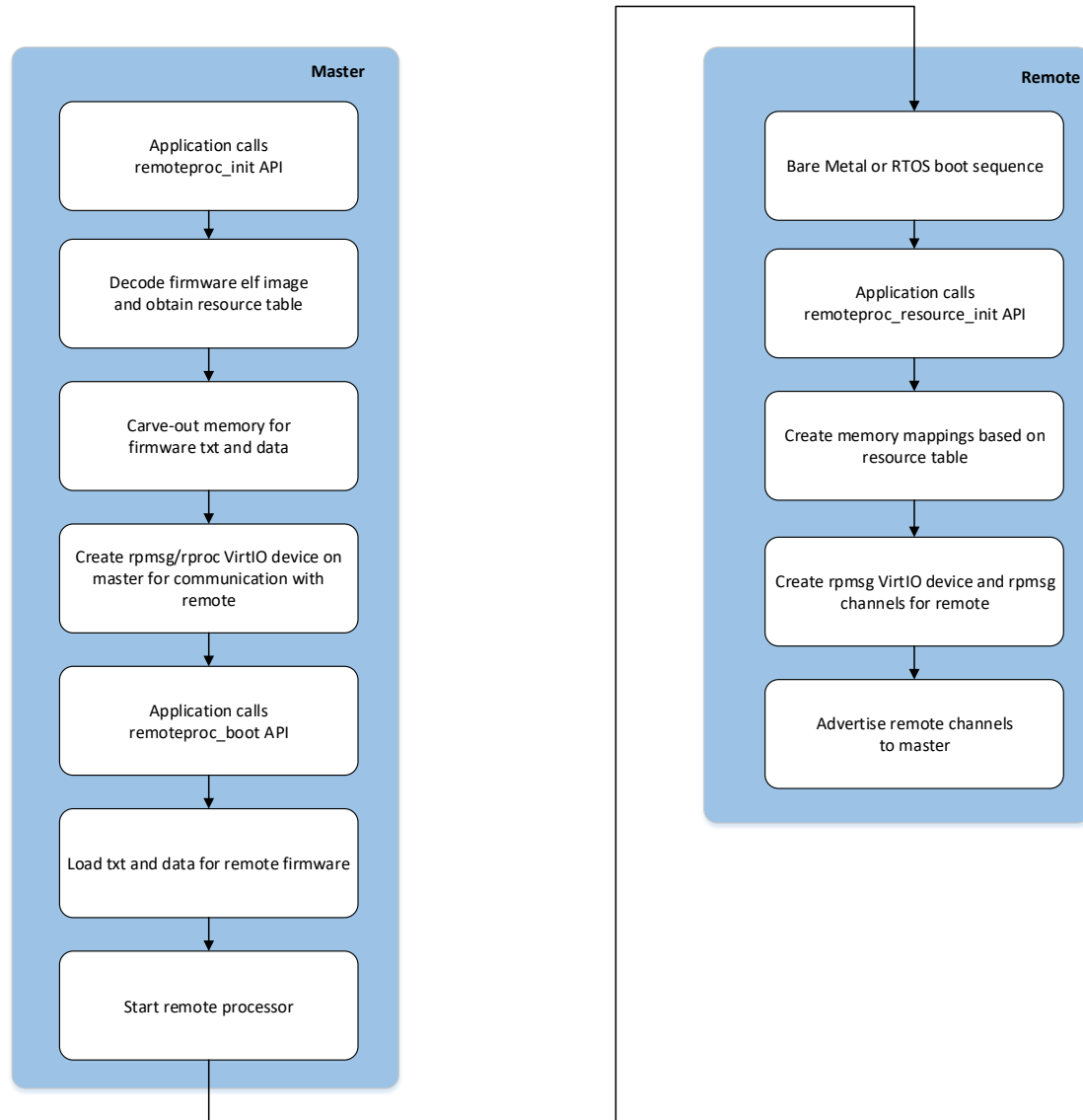


Figure 1. Bootup with remoteproc: Conceptual Diagram

2.2. Software Environment or Configurations For OPENAMP

The OpenAMP Framework can be used with RTOS or bare metal contexts on a remote processor to communicate with Linux applications (in kernel space or user space) or other RTOS/bare metal-based applications running on the master processor through the remoteproc and RPMsg components. The architecture can be used in Data Intensive Application where more focus is on display interfaces in non-critical subsystems. The architecture can be tweaked to work in safety-critical systems. The OpenAMP Framework also serves as a stand-alone library that enables RTOS and bare metal applications on a master processor to manage the life cycle of remote processor/firmware and communicate with them using RPMs

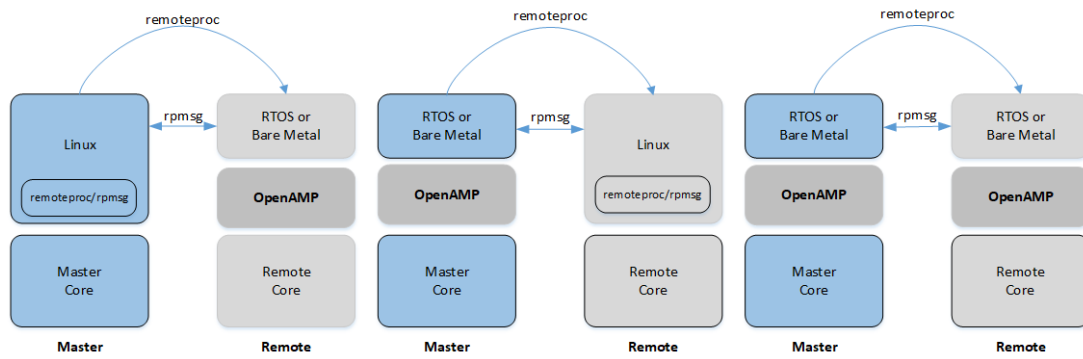


Figure 2: Software Environments/Configuration for OpenAMP

3. DESIGN FOR MIXED SAFETY CRITICAL SYSTEMS

The main requirement for a safety-critical system is the isolation to separate different software components from each other. Today, several hardware-assisted separation capabilities are provided by various MPSOC manufacturers to isolate safe and non-safe domain. The separation is required for the processing blocks, memory blocks, peripherals, and system functions. The OpenAMP framework helps facilitate the implementation of fault-tolerant systems. The OpenAMP framework can enable an RTOS on the safety critical processor, which is the system master and manage the critical system operations, to control the lifecycle on the application processor. If there is a failure, the RTOS can simply reboot the system without impacting the operation of the rest of the system.

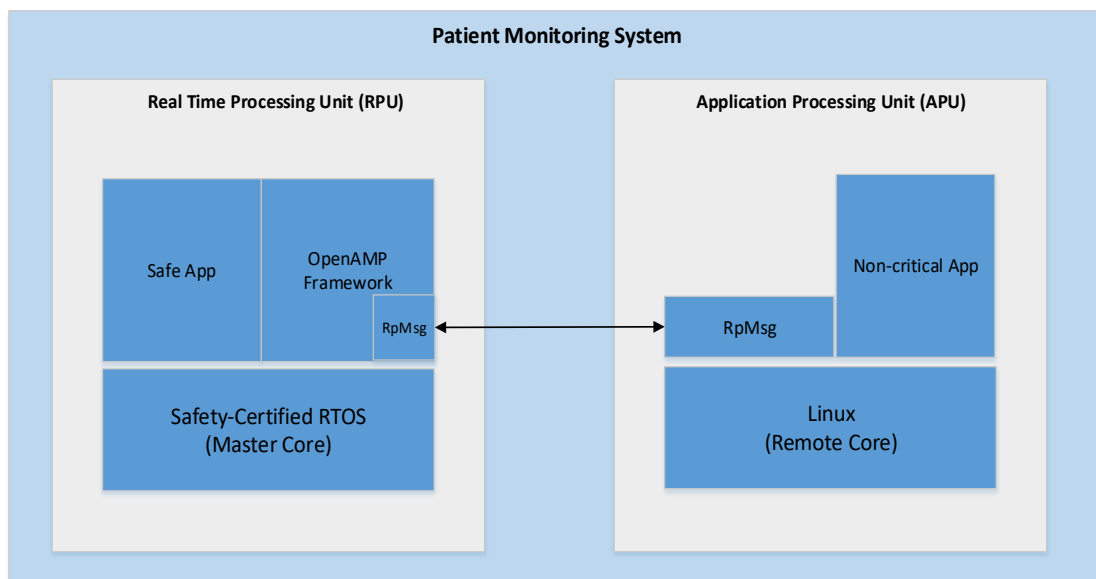


Figure 3. Example architecture diagram illustrating a mix-critical safety system in Patient Monitoring System

Let us take the example of a patient monitoring system (Figure 2), with a platform that comprises a cluster of ARM Cortex A5 as a Real-time Processing Unit (RPU) and ARM Cortex-A53 core as an Application Processing Unit (APU). The functional safety-certified software context RPU obtains the sensor data. The sensor data contains the patient vitals. The RPU main purpose is to monitor the patient vitals. The non-critical subsystem in the APU

consists of the high-level operating system, which displays the data on an LCD and provides internet connectivity. The OpenAMP helps to isolate the critical from non-critical subsystems and communicate person vitals from RPU to APU using OpenAMP RMPMsgs.

3.1. Hybrid Design Approach

The current trend is to merge both hypervisor and OpenAMP framework to form a hybrid design. This helps solve problems related to mixed critical systems in various applications areas across automotive, medical, banking, and many others.

In modern automotive applications like Digital cockpit, there is a need to separate safety critical from the non-critical components. To address such a situation, a hybrid design (Figure 3) can be used where both the advantages of hypervisor and OpenAMP framework can be combined into the architecture to create a zone of trust for the communication where data exchange can happen safely between remote and master.

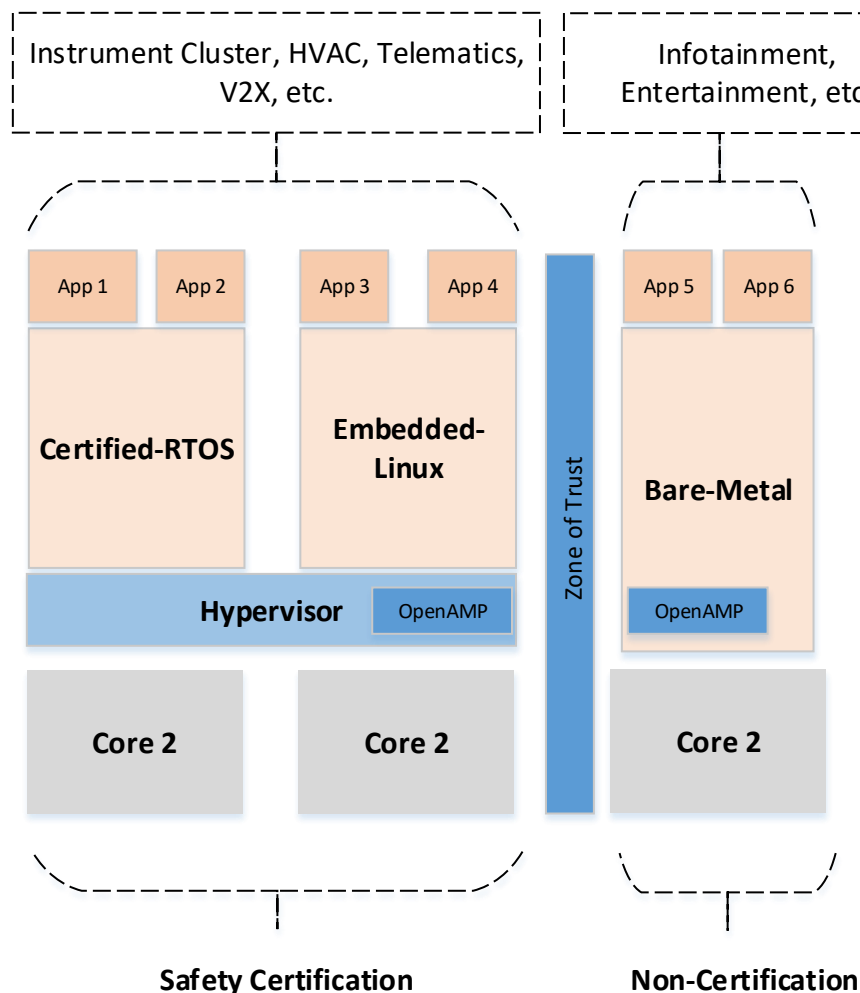


Figure 4. Hybrid design approach in a Digital Cockpit

4. CONCLUSION

The multi-core processor SoCs can deliver an enhanced level of unified peripheral and CPU clusters which enable engineers to lower BOM costs, reduce power consumption and accelerate design implementation. Multicore designs are becoming increasingly common and years to come will be mainstreamed. With the increase of multicore design, we will see rise in usage of OpenAMP. It is worth mentioning that OpenAMP has indeed opened up many new doors to relook at the mixed-safety critical applications. OpenAMP is a platform to stay and provide a solution to implement mixed safety-critical solutions in a robust and cost-effective manner.

ACKNOWLEDGEMENTS

The author would like to thank my organization L&T Technology Services for giving me opportunity to work on OpenAMP. Also like to thank the OpenAMP open source community who is trying to help and contribute to this innovative software framework. OpenAMP as an emerging API standard managed under the umbrella of Multicore Association. This project is jointly maintained by Mentor Graphics, Xilinx and other software and hardware vendors. A current reference implementation of the proposed OpenAMP standard is available at: <https://github.com/OpenAMP/open-amp>. Mentor Embedded Multicore Framework (MEMF) is a proprietary implementation of the OpenAMP standard.

REFERENCES

- [1] OpenAMP GitHub: <https://github.com/OpenAMP/open-amp>
- [2] OpenAMP Project: <https://www.openampproject.org/>
- [3] Mentor Graphics, OpenAMP Framework User Reference.pdf
- [4] Xilinx, Libmetal and OpenAMP User Guide
- [5] OpenAMP Wiki: <http://www.wiki.xilinx.com/OpenAMP>
- [6] Xilinx OpenAMP source code: <https://github.com/Xilinx/open-amp>
- [7] Embedded Computing Design: Multicore and OpenAMP:
<https://www.embeddedcomputing.com/technology/processing/compute-modules/multicore-and-openamp>

AUTHORS

Mridula Prakash has over 13 years' experience in the electronics industry, largely dedicated to embedded software. She is a senior executive with extensive experience in architecting and building embedded products in the Industrial Products domain. She is an active member in embedded system community and has been working on various technologies of microprocessors and microcontrollers, including x86, PIC, AVR, MIPS, PowerPC and ARM, developing firmware and low-level software in C/C++ on Linux, Android, FreeRTOS and many other kernel and operating systems.



In her current role as Specialist – Embedded Architect at L&T Technology Services (LTTS), she is responsible to understand major trends in the embedded sector and help in the implementation of embedded design software and modernization of legacy systems.

KEY LEARNINGS FROM PRE-SILICON SAFETY COMPLIANT BOOTROM FIRMWARE DEVELOPMENT

Chidambaram Baskaran, Pawan Nayak, R.Manoj,
Sampath Shantanu and Karuppiiah Aravindhnan

Texas Instruments India Ltd, Bangalore, India

ABSTRACT

Safety needs of real-time embedded devices are becoming a must in automotive and industrial markets. The BootROM firmware being part of the device drives the need for the firmware to adhere to required safety standards for these end markers. Most software practices for safety compliance assume that software development is carried out once the devices are available. The BootROM firmware development discussed in this paper involves meeting safety compliance need while device on which it is to be executed is being designed concurrently. In this case, the firmware development is done primarily on pre-silicon development environments which are slow and developers have limited access. These aspects present a unique challenge to developing safety compliant BootROM firmware. Hence, it is important to understand the challenges and identify the right methodology for ensuring that the firmware meets the safety compliance with right level of efficiency. The authors in this paper share their learnings from three safety compliant BootROM firmware development and propose an iterative development flow including safety artefacts generation iteratively. Concurrent firmware development along with device design may sound risky for iterative development and one may wonder it may lead to more effort but the learnings suggests that iterative development is ideal. All the three BootROM firmware development has so far not resulted in any critical bugs that needed another update of the firmware and refabrication of the device.

KEYWORDS

Concurrent development, Firmware development, Safety compliance, Pre-silicon software development.

1. INTRODUCTION

The challenges to coordinate the different product developments activities have dramatically increased with Concurrent Engineering and Integrated Product and Process Development [1]. In order to accelerate the time to market, firms attempt to overlap the different activities in product design and development – leading to iterative overlapped development. Safety software development has typically followed the traditional highly-structured approaches such as V-model or waterfall [2]. The V-model [3] is composed of well-defined 9 steps through project initiation, design, test, maintenance and phase-out. A recent study of the safety software development and agile development [4] indicates that the agile methods have been not adopted significantly. When there is a need to adopt these methods to specific safety software development with constraints such as concurrent development and limited access to test environments, there is not much study done that can be beneficial and reused. Authors in this paper attempt to provide few key learnings

from safety firmware development concurrent with hardware design in a constrained pre-silicon environment and show efficient ways to meet the safety compliance.

2. SAFETY SOFTWARE DEVELOPMENT

Automotive industry has adopted usage of electronic control units (ECUs) in a large scale within a very short period of time. Large number of processors are heartbeats of these ECUs and they perform several safety critical functions [5]. One of the key requirements for processors or devices being used in these functions is for firmware in the ROM of these devices to be safety compliant. The most popular standard for safety compliance is the ISO 26262 standard titled “Road vehicles — functional safety” [6]. The compliance to this standard needs’ adoption of software practices and tools that to demonstrate the compliance to standard and ensuring quality of the software. This needs compliance across OEMs, their suppliers, and developers of automotive components. Part 6 of this standard [6] details the practices to be adopted by software developers. The standard requires well documented and detailed requirements followed by design details documentation and finally good test plans. These artefacts need to be thoroughly reviewed and also traceability of the requirements to design to test is critical to ensure quality of the delivered software. It is very essential to prove that the development meets the compliance requirements. Addition to the detailing the implementation aspects, compliance to coding standards though MISRA-C [7] and dynamic coverage of the code through testing is also mandatory. The final resulting firmware must be well tested and test results produced to show that the firmware has zero possible bugs. Most of the literature details methodology and practices for safety software development that is significantly different from the pre-silicon firmware development presented by the authors in this paper. The authors discuss about safety compliance for firmware development while the device is being designed.

3. PRE-SILICON SOFTWARE DEVELOPMENT CONSTRAINTS

The firmware development discussed in this paper involves concurrent development while the device on which it is to be run is still being designed. This concurrent development of the device and the firmware enables shorter time to market as the firmware is put into the ROM as soon as the device design is completed and hence built into the fabricated device ROM. However, this poses several challenges in terms of availability of testing platform for firmware development as the actual device is still being designed. The testing platforms used for these scenarios are referred to as pre-silicon testing platforms. Several challenges of pre-silicon testing platform are listed below.

3.1. Speed of the pre-silicon Platform

Software developed pre-silicon needs a testing environment to test to ensure it has near zero defects. These test environments are very slow since the entire design of the device is emulated using another hardware. For example, for the devices for which the authors have developed the firmware these environments run at 100 KHz while the real device can run at close to 100 MHz. This slowness has a direct impact on the amount of time spent on testing. For example, for the firmware development needed almost 10 days of testing time due to the slowness of the environment.

3.2. Cost of the pre-silicon Platform and access time

The pre-silicon platform is very costly and typically only couple of platforms are available for each device design. These platforms are used by multiple teams due to hardware-software

concurrent development and hence different teams are provided a very short period of access to these platforms. For instance, the firmware team of 3 software engineers in total had access time of 40 hours per week – approximately 14 hours per week per engineer. Safety compliance needs several test results and artefacts to be generated and hence the slowness of the testing environment presents a unique challenge.

4. CONCURRENT DEVELOPMENT CHALLENGES

The authors worked on firmware development while the design of the device on which the firmware is expected to run was also being developed concurrently. This type of concurrent development introduces additional challenges to the safety compliance for firmware development.

4.1. Out of sequencing of features development

In this type of concurrent development - some of the features of the device may be available towards end of the hardware design and hence software team will have to develop these features without having any platform to validate them since the testing environment is built from the completed hardware design. This results in quite a bit of time gap between completion of design, implementation and testing of the firmware.

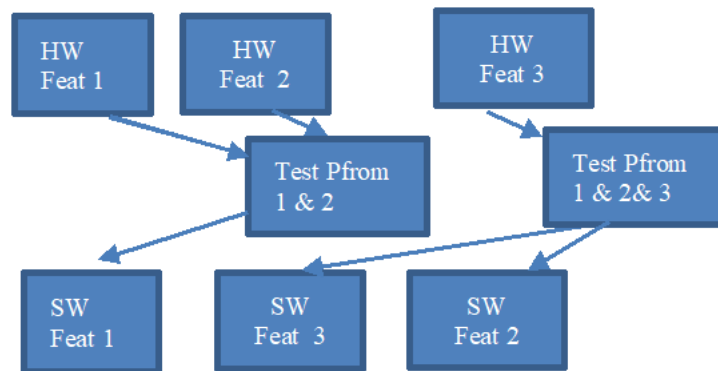


Figure 1. Out of sequencing of HW and SW features

Figure 1 shows timeline sequence of a scenario where the device hardware features implementation is in a different sequence compared to firmware features. This can happen as the effort to design the hardware feature and the related firmware feature may not be very similar and also team sizes working on these can differ. Due to these reasons out of sequence of development was found to be very common in all the 3 firmware projects. As a result, the test platforms for firmware testing may be available at a later point in time well beyond the implementation of the firmware.

4.2. Cross functional team bandwidth for reviews

Firmware software is usually reviewed by teams that are also involved in design of the device and the testing of the device. Many aspects of the firmware also pertain to aspects like device qualification, device characterization and hence the firmware design and implementation needs to be carefully reviewed by cross-functional teams. As the different teams involved in the device development concurrently, the availability of different team members for reviews is a challenge. For safety compliance it is important to review the design, implementation, test plans and test

results at the right time with right level of rigor. The reviews need to be recorded and quality of the reviews have to be met.

5. KEY LEARNINGS

Most of the literature discuss the challenges in meeting safety compliance in software development that is typically carried out on a platform where the final device on which this software needs to run is already available (referred to as post-silicon software development). Development of safety compliant firmware while the device itself is being designed is very special case which opens up new challenges. Authors in this paper discuss the key learnings from three such firmware development projects. The understanding of the constraints of the development environment, concurrent development and safety compliance challenges can enable in efficient and repeatable methodology for pre-silicon safety compliant firmware development

5.1. Safety process challenges for pre-silicon safety compliance in concurrent development

In this type of concurrent development - some of the features of the device may be available towards end of the hardware design and hence firmware team will have to develop these features without having any platform to validate them since the testing environment is built from the completed hardware design. This results in quite a bit of time gap between completion of design, implementation and testing. Authors in their first firmware development found that during the critical phases of the design and implementation cross functional teams were also nearing completion of their milestones leading to time constraints. This resulted in delays in reviews and feedback which are very essential for safety compliance. **Learning 1** – Ensuring the cross-functional team plans are well synchronized on a periodic basis and not just at the beginning of planning for dependencies on deliverables but also at completion of design feature wise helped in streamlining the development. The traditional firmware development focused on completing the entire design and then focused on reviews but the authors soon found out that each feature level review was more productive from better reviews as well as planning perspective.

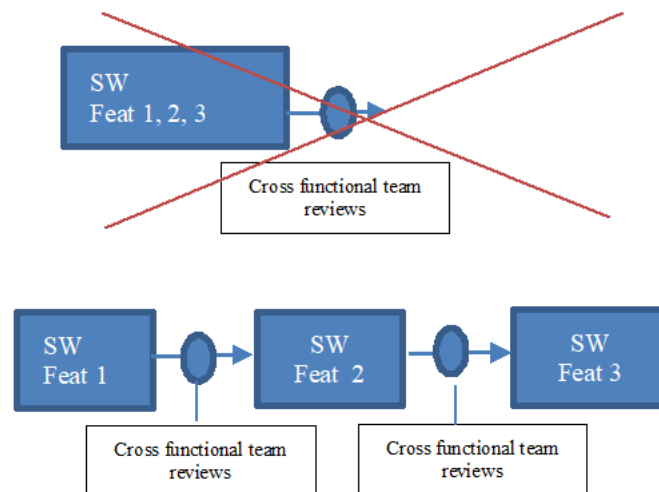


Figure 2. Balancing the reviews – periodic reviews

It is recommended that the availability of team members especially multi-functional team members who are also involved actively in their own domain deliverables is available for

reviews. Cross functional teams bring more insight into the design - the architect of the device has good overview of the usage requirements of a customer, while the team that is involved in device characterization can provide inputs on testing aspects. **Learning 2** – The sequence of the reviews is also very critical. Typically, firmware adds few new features while most of the other features are reused from prior devices. Focusing on the new features early - design review, test plan reviews enabled effective reviews early, better quality of the design and also provided sufficient inputs to improve the implementation for safety compliance. An incremental review process with new features being reviewed early has been found to be very effective. Interestingly these new features need to be reviewed also towards end of device design as the other teams involved in the design would have learnt a lot more of the details as well. This is a very unique review flow that authors identified to be effective in firmware development that is carried out concurrently with device design.

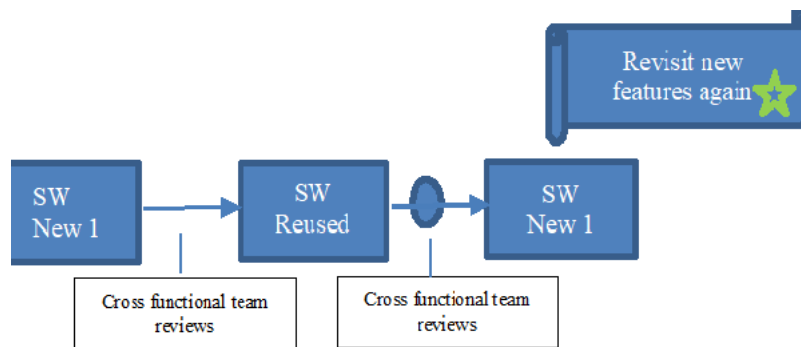


Figure 3. New features early and revisit reviews again at end

5.2. Safety process challenges for pre-silicon safety compliance in concurrent development – artefacts generation

Several artefacts need to be generated for safety compliance. It is important to understand the limitation of the testing environment and speed so that the generation of these artefacts can be planned better. **Learning 3** – The time for artefacts generation were overlooked in the first firmware development. The initial thought process was to generate some of the artefacts like code coverage report towards end of the firmware development so that final reports needed for meeting safety compliance can be made available. The time taken to generate these dynamic analysis report almost took 1.5x of the total testing time as the testing environment was not available continuously and the tests had to run and re-run to generate for any coverage gaps. Authors recommend that these dynamic analysis report generation be done module wise as and when they are completed to look for any code coverage gaps. This not only shortens the time to run (since it is done at a smaller module level) but also to quickly address the gaps to generate new tests to run.

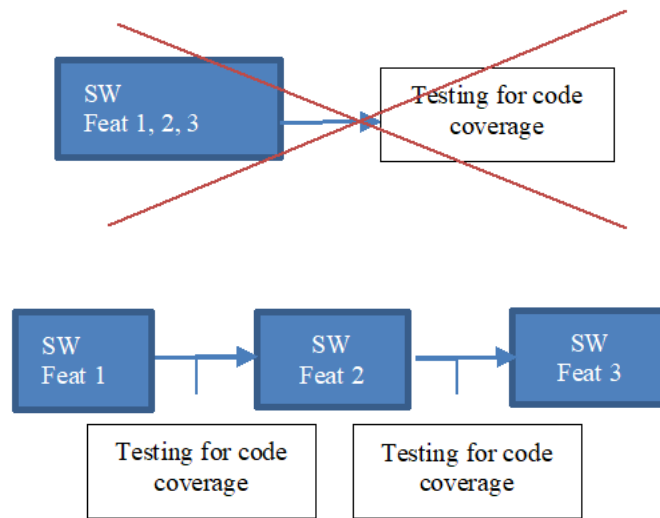


Figure 4. Dynamic analysis – code coverage iteratively

Authors recommend that the artefacts needed for safety compliance be classified into 2 classes – one that needs the test environment and one that is static – without needing any test environment. The artefacts that need the testing environment needs to be planned well for pre-silicon firmware development. It is highly recommended that these artefacts if they can be generated incrementally through the development cycle then they need to be generated periodically.

- Static artefacts – requirements, design document, test plan, traceability from requirements to design to test, MISRA-C compliance report
- Dynamic – needing test environment – test results, dynamic analysis

The dynamic analysis report (code coverage from testing) generation is heavily dependent on the testing environment and hence it is a key item to be planned well ahead. Each team member does not get a continuous access to the pre-silicon development environment and hence the generation of tests for coverage has to be planned well. In a post-silicon software development this is a not a key constraint as the environment to test is always available and each member may have exclusive test setup.

5.3. Safety process challenges for firmware code that is reused from non-safety development

The firmware development is usually is not written from scratch and multiple parts of the firmware is reused from older devices as well. One of the challenges in this reuse is that those reused pieces of firmware may not have gone through the safety compliance needs. Authors in their firmware development had significant portions of reused software and identified several artefacts that can help in identifying the quality of these reused software through mapping the functionality of reused software to safety features expected and identifying the level of rigor needed for safety compliance. **Learning 4** - The pieces of firmware that needed rigor was found to be portion of software that is involved in configuring registers in the device that can cause the functionality failure at run-time. Focusing on these aspects enabled building the rigor for the reused firmware pieces. The start-up booting time failures were made to return error values that can be handled at the application level and hence less rigor was needed for these failures. Further the reused firmware features were covered 100% with tests and traceability reports were

generated to ensure that these were fully tested. Dynamic analysis coverage was also another aspect added to ensure that the coverage of the reused firmware was close to 100%. These efforts saved significant time without having to go through code reviews and design reviews of the several thousand lines of code that were reused.

6. RECOMMENDATION FOR ITERATIVE DEVELOPMENT

The authors through their learnings from three safety compliant BootROM firmware development projects recommend that the development must be carried out iteratively. Concurrent development along with device design and development may sound risky for iterative development and one may wonder it may lead to more effort but the learnings suggests that iterative development is ideal.

- Suggested methodology is to first start with new features, complete and then move to reused or known features. The iterative development with new features designed, reviewed, tested followed by reused features ensures review rigor and early identification of problems.
- Iterative generation of safety collaterals - Iteratively generate the safety collaterals like design document, test cases and also generate reports from testing like dynamic code coverage through the feature development given the pre-silicon environment challenges.
- Revisit the new features design, test cases one more time towards end of the firmware development to look for newer understanding from the cross-functional teams as those teams also would have completed their implementations and tests for the new features. Several new findings and improvements were seen during the second round of reviews.

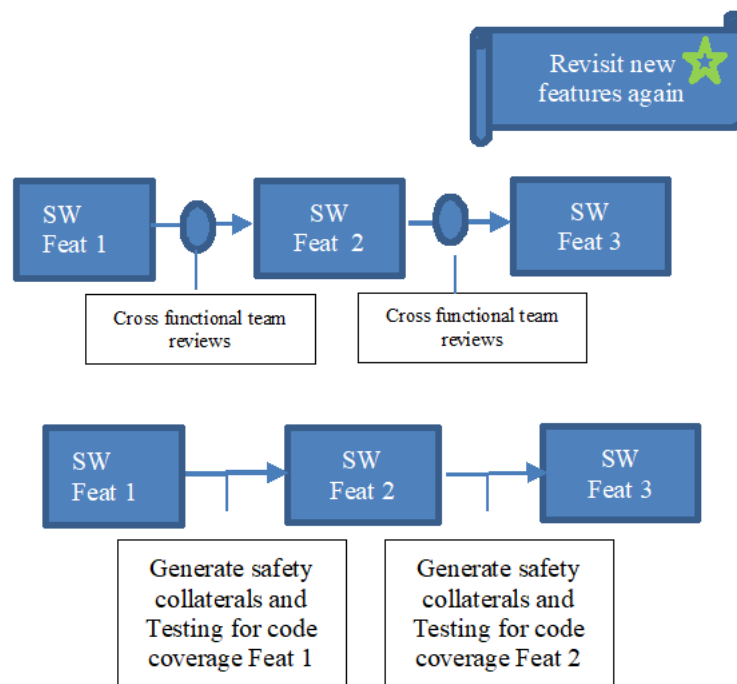


Figure 5. Iterative development flow

7. CONCLUSION

In order to accelerate the time to market, firms attempt to overlap the different activities in product design and development – leading to iterative overlapped development. Authors in this paper present learnings from such a development where there were additional challenges in needing to develop firmware concurrently with device design along with the limitation of pre-silicon platform. Interestingly it was observed that the iterative development of the firmware through new features first and then towards reused features provided optimal usage of time and effort. The constraints of the pre-silicon environment pushed for early test reports generation in an incremental manner so that the environment could be used efficiently. It was also found that it is essential to revisit the design, testing of the new features at the end of the firmware development to incorporate any new learnings from cross-functional teams as these teams also would have learnt from their own work. The synchronization of design and testing is a huge challenge due to different team sizes and efforts and hence ensuring one final design and test review when cross-functional teams have also progressed helped in identifying errors and solidifying the new features in firmware. This is another unique aspect recommended by the authors.

REFERENCES

- [1] Browning, T.R., Eppinger, S.D., 2002. “Modeling impacts of process architecture on cost and schedule risk in product development”. *IEEE Transactions on Engineering Management* 49 (4), 428–442.
- [2] Roger S. Pressman, “Software Engineering: A Practitioner’s Approach 7th Ed”, MacGrawHill, p.40-41, 2010.
- [3] Paul Rook, “Controlling software projects”, *IEEE Software Engineering Journal*, vol. 1, no. 1, p.7-16, 1986.
- [4] Rashidah Kasauli, Eric Knauss, Benjamin Kanagwa, Agneta Nilsson and Gul Calikli Chalmers “Critical Systems and Agile Development: A Mapping Study”, 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, 470-477.
- [5] Georg Georgakos, Ulf Schlichtmann, Reinhard Schneider, and Samarjit Chakraborty. “Reliability challenges for electric vehicles: from devices to architecture and systems software” In *Proceedings of the 50th Annual Design Automation Conference*, page 98. ACM, 2013.
- [6] ISO 26262-6:2018 Road vehicles — Functional safety — Part 6: Product development at the software level.
- [7] Motor Industry Software Reliability Association et al. MISRA-C: 2004: Guidelines for the Use of the C Language in Critical Systems.
- [8] Xiaocheng Ge, Richard F Paige, and John A McDermid, “An iterative approach for development of safety-critical software and safety arguments”, In *Proc. of AGILE Conf.*, pages 35–43, Nashville, TN, USA, 2010. IEEE.

AUTHORS

Chidambaram Baskaran, Pawan Nayak R.Manoj, Sampath Shantanu and Karuppiah Aravindhan are part of the Texas Instruments India (Ltd) with key areas of interest being embedded software development, ROM development and driver development for peripherals.

TOWARDS MAINTAINABLE PLATFORM SOFTWARE - DELIVERY COST CONTROL IN CONTINUOUS SOFTWARE DEVELOPMENT

Ning Luo and Yue Xiong

Visual Computing Group,
Intel Asia-Pacific Research & Development Ltd, Shanghai, China

ABSTRACT

Modern platform software delivery cost increases rapidly as it usually needs to align with many hardware and silicon's TTMs, feature evolution and involves hundreds of engineers. In this paper, citing one ultra-large-scale software - Intel Media Driver as an example, we analyse the hotspots leading to delivery cost increase in continuous software development, the challenges on our software design and our experiences on software delivery cost shrink against the targeted design enhancements. We expect the identified hotspots can help more researchers to form the corresponding research agendas and the experiences shared can help following practitioners to apply similar enhancements.

KEYWORDS

Software Delivery Cost Control, Predictable Software Evolution, Streamlined Parallel Development, Continuous Integration.

1. INTRODUCTION

Modern platform software delivery cost increases rapidly as it usually needs to align with many hardware and silicon's TTM, feature evolution and involves hundreds of engineers. In this paper, citing one ultra-large-scale software - Intel Media Driver as one example, we analyse the hotspots in continuous software development leading to delivery cost increase, the corresponding software design challenges and our experiences on software delivery cost shrink by targeted design enhancements. We expect the identified hotspots can help more researchers to form the corresponding research agendas and the experiences sharing can help following practitioners to apply similar enhancements.

2. DELIVERY COST IN CONTINUOUS DEVELOPMENT - HOTSPOTS AND CHALLENGES

Intel Media Driver is an ultra-large-scale platform software with around 3 million lines of code and supported by over 300 developers. As the bridge between Intel GPU (graphics processing unit) and the ever-changing end to end media usages, Intel media driver is designed for multiple generations' Intel GPU support on top of different OS and API. It is widely used in diverse media usages ranging from client to cloud, against different software stacks. Every year, it has over one hundred software releases for different purposes.

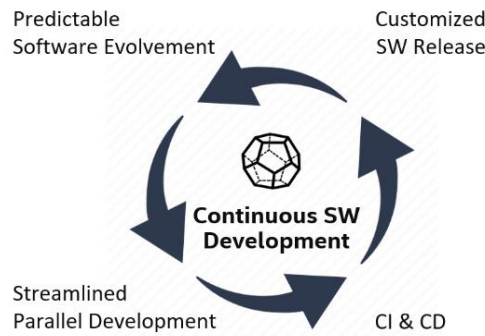


Figure 1. Delivery Cost in Software Development Cycle

The software delivery cost for Intel Media driver is introduced from various aspects of the continuous software development, as can be shown by Figure 1 above.

2.1. Predictable Software Evolvment

Execution Predictability is crucial to the long-term success of the large-scale platform software which usually includes the support to dozens of different hardware products.

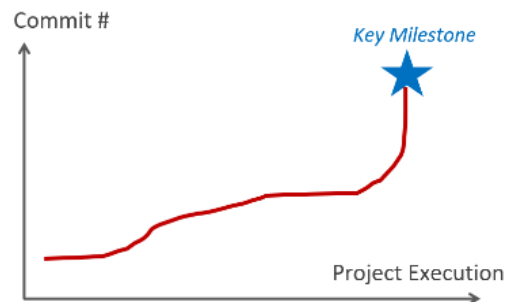


Figure 2. Predictability Gap in Software Development

As can be shown by Figure 2 above, in the new product enabling, lack of development predictability will lead to extensive rush high risk changes toward cycle's end and frequent product KPIs delay. To those software projects with multiple products enabled/maintained in parallel, the risk will be further amplified because of the unexpected interactions. Finally, we will have to face with more and more unplanned tasks and exponentially increased software delivery cost.

Many factors can lead to the predictability decline in software evolvment. Design perspective, it can be connected to the keep increasing software complexity and gaps on code and effort reusing. Taking Intel Media driver as one example, its software complexity mainly comes from three perspectives –

- **Hardware Complexity**

Intel's GPU portfolio has grown immensely over the last decades. Media driver now needs to support over 50 different media hardware sub-engine IP (Intellectual Property) cores and over 20 different media SOC's (system on chip).

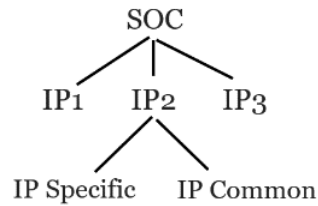


Figure 3. Software Abstraction for Intel GPU

As can be shown by Figure 3. above, Sub-engine IP is the basic hardware unit for Intel GPU. Software abstraction for IP is responsible for GPU workload generation on specific sub-engine. It is the most complex and expensive piece in our new product enabling. Previously, to enable a new version sub-engine IP, we need totally rewrite the corresponding logic, although significant portions of the IP logic is common and can be reused across different IP versions. E.g., the hardware unrelated logic and the hardware features with no changes. It causes not only bigger enabling effort, but also potential risks on the common part divergence between different IPs.

Media SOC is composed of one set of different Media IPs against the SOC specific customizations. For SOC's belonging to the same product family, more than 90% of the logic are common and embodied in the underlying sub-engine IPs. Previously, to enable a new SOC, almost all underlying IPs will be impacted (for the SOC specific programming), while those changes inside underlying IPs can further impact other SOC's belonging to the same product family (based on the same sub-engine IP set). Finally, it will lead to huge code changes impacting hundreds of files together and unexpected big enabling and maintenance effort.

- **OS/API complexity**

Intel Media driver now supports 4 different OSes and 4 different APIs. Although over 85% logic of Media driver is shared between different OS/API, the proportion of the code and effort sharing is much lower. E.g., Changes deemed as OS/ API specific can cause unexpected impacts on other OS/APIs occasionally, so duplicated validation effort seems inevitable.

- **Usage Complexity**

Intel Media driver is widely used in diverse media usages ranging from client to cloud, like Video Playback, Streaming, Gaming, Conference, media delivery and media analytics. For each usage, we need the customized settings on hundreds of different hardware knobs. Without good decoupling, usage specific customizations can easily be mixed up and lead to big debugging and maintenance effort.

In summary, the keep increasing software complexity can ruin the predictability in continuous software development. To mitigate the risk, it requires better code/effort reusing and less mutual impact between software modules in different dimensions - SOC's, IPs, Features, OS/API and Usages.

2.2. Streamlined Parallel Development

Modern platform software often need face with the challenges from parallel development on multiple development paths. Intel Media Driver is one such example. Partial of its source code (mainly for published products' support on Linux) is open sourced and open for contributions from community. While the rest part is kept close source for internal development only. Parallel development is required on both paths. In execution, parallel development can lead to duplicated

effort on code change preparation, code review, static analysis, and validation. Even worse, in the long run, code divergence between two code paths turns to be inevitable and the maintenance effort will be multiplied.

To mitigate the risk, it requires a streamlined development model on top of “single code base”.

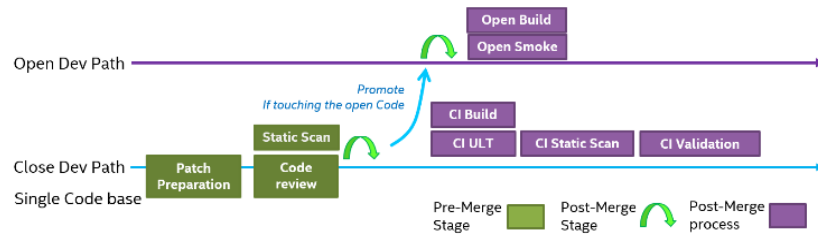


Figure 4. Streamlined Parallel Development

The Figure 4 above demonstrates the streamlined development model we used in Intel Media Driver development. The “single code base” will be directly used for internal development while open-source development will be based on its open-source-able subset. After open-source-able subset derived from the single code base, all internal code changes (accounting for around 98% of the total commits) only need to be merged into the “single code base”, with one time patch preparation, code review, static code analysis and validation. The corresponding patches to the open-source-able code base will be automatically generated after stripping out the internal only parts. While for the left 2% code commits contributed by community on open-source code base, they will be automatically ported back to the “single code base” for internal review and validation before code merge. In this way, we can best automate and streamline the development flow to minimize the duplicated effort and code divergence on two development paths.

However, to support the streamlined development model, it requires one well organized source code architecture fully aligned with the open-source requirement so that the build system can easily strip out the non-opensource-able part in each code commit with minimal conflict and human intervention.

2.3. Continuous Integration & Continuous Delivery

Continuous Integration (CI) and Continuous Delivery (CD) are the software engineering approaches to promise the reliable software release at any time. They are triggered by each code commit and include series of automatic build and automatic validation stages.

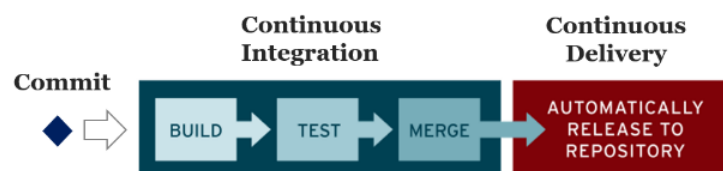


Figure 5. Continuous Integration (CI) and Delivery (CD)

To key to a successful CI/CD system is the quick enough developer feedback loop against sufficient build and validation coverage. Still taking Intel Media driver as one example, on average there are around 15 commits per day. It means to promise the best CI efficiency, the CI build and validation stages for each commit need be finished in less than 2 hours. While in real

case, because of the large scale of the software, each media driver build needs to include 4 binaries and the corresponding build and Validation stages require around 5 hours. Obviously, build and validation acceleration is critical for the effective CI/CD.

- **Build Acceleration**

Modular build can help to accelerate the build stage. Against modular build, each build request can be broken into smaller build tasks based on sub-modules. In a distributed CI build system, those sub-tasks can be dispatched to multiple machines for parallel build. Meanwhile, some intermediate build results (object files or static libraries) can be reused between several correlated build tasks. Against modular build can help us achieve 3 to 4 times' build time shrink.

But to support modular build, design perspective, our software need be fully decoupled into independent sub-modules.

- **Validation Acceleration**

Smart validation can help the validation stage acceleration which is essentially to choose the just enough validation coverage for each commit based on the code change.

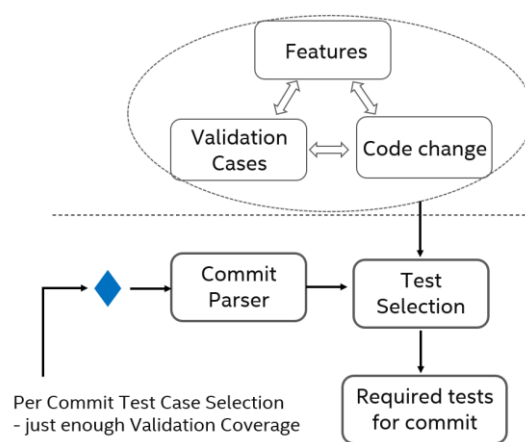


Figure 6: Smart Validation Working Flow

Smart Validation infrastructure for Intel Media driver can be shown as Figure6 above. The basis of the smart validation is the traceability from source code to the impacted features and the related test cases. Against the traceability, for each new commit, validation system precisely can easily deduce the most proper validation coverage from the code change. Design perspective, it requires a software design with fully decoupled software features.

2.4. Customized Software Release

Customized Software Release is essential to large scale software which is the basis for software offering differentiation based on customer needs and IP exposure control on unpublished modules. To support the customized software release. It requires a flexible enough software design for on-demand software tailoring with minimal impact. But to a complex platform software like Intel Media driver, without good decoupling, SOC/IP/feature-based tailoring can easily lead to big changes impacting hundreds of files together. Even worse, the big change can trigger unexpected regressions on unrelated software modules which will lead to bigger validation,

debugging and maintenance effort. To fix the gap, design perspective, it requires fully decoupled software modules for different OSes, APIs, hardware platforms, and functional domains.

3. OPPORTUNITIES - DESIGN ENHANCEMENT

Effective software delivery Cost Control cannot work without a good software design. We may still take Intel Media driver as one example to analyse how the targeted design enhancements can help on the delivery cost shrink.

Based on the above analysis, we try to have better software decoupling in various dimensions, as can be shown by Figure 7 below.

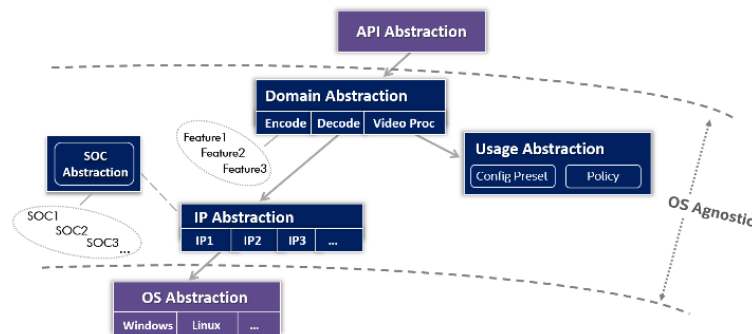


Figure 7. Enhanced Software Design

3.1. OS and API

In Intel Media driver, only around 15% logic is OS dependent. Two OS dependent modules are introduced to abstract those OS and API differences. At the top, API abstraction module will translate different industry standard APIs into one set of uniform OS agnostic business API. At the bottom, OS abstraction module will abstract services exposed by different OS/KMD into the uniform OS service APIs. With the above two OS dependent modules, the enabling and maintenance effort on the ~85% OS agnostic parts can be better shared across various OS/API. Meanwhile OS/API based source code tailoring can be easily applied against file/folders. It can greatly benefit our parallel development, modular build, and customized software release.

3.2. Hardware

- **Abstraction for Sub-engines**

The software abstraction for media sub-engine is decoupled into two layers: One is the hardware unrelated domain abstraction modules mainly responsible for industry standard translation. They can be shared by different sub-engine IPs during our hardware evolvement. The other is hardware related IP abstraction modules responsible for hardware related programming. It will be our focus in the new sub-engine IP enabling. By separating domain abstraction layer from IP abstraction layer, it greatly reduces our enabling and maintenance effort on the around 40% hardware unrelated logic.

- **Abstraction for Features**

The Domain abstraction module and GPU abstraction module are further decoupled into different hardware features and the logic specific to each feature is centralized maintained in the same code block. Feature wise abstraction can help us achieve better traceability from usages/features to source code and greatly benefits the parallel development and smart validation.

- **Abstraction for SOC**s

The SOC specific customizations will be decoupled from the underlying IP and centralized maintained in dedicate SOC abstraction modules. Different SOC's will be fully isolated by files. In this way, to enable a new SOC, the files impacted can be greatly decreased from over 100 to less than 5 which will greatly reduce our enabling and maintenance effort on dozens of different SOC's.

3.3. Usage

Usage based customizations is centralized maintained in usage abstraction modules. Each usage has its own abstraction and different usages will be fully isolated in specific files. In this way, it can avoid interactions between various usages, greatly facilitate the usage specific debugging/tuning and reduce the maintenance effort.

4. RESULTS

Combining the aforementioned design enhancements together, it dramatically boosts our engineering efficiency and shrinks the software delivery cost, which can be demonstrated by various engineering measurements –

- The proportion of code commits happened in the last 2 weeks before the important product milestones fell to 8 percent from a 2018 high of 18 percent.
- Average time span for a new commit to be merged into both development paths (open source and close source) fell to 3 days from 4 weeks.
- Average CI developer feedback loop for each code commit was decreased from 5 hours to 1.5 hour.
- Average turnaround time for one new customized software release requirement was reduced from 3 weeks to 2 days.

In total, with the same number of developers, now the team can support 2.5 times more SOC's and IP's comparing with 2018.

5. CONCLUSIONS

Delivery cost control is crucial to the success of large-scale platform software. We expect the identified hotspots above can help more researchers to form the corresponding research agendas and the experience shared can provide follow practitioner with insights on similar enhancements.

ACKNOWLEDGEMENTS

Thanks to all colleagues working on refactoring for continuous software delivery and competency improvement. Appreciate your hard work to turn all our good designs into the reality.

REFERENCES

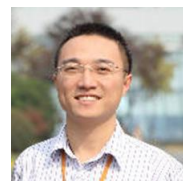
- [1] Martin Fowler, “Refactoring–Improving the design of Existing code ”
- [2] Martin Fowler, “Refactoring Guideline ”, <https://refactoring.com/>
- [3] Lianping Chen, “Continuous Delivery: Huge Benefits, but Challenges Too”, Publication Year: 2015, Page(s):50 – 54, IEEE Software
- [4] J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley Professional, 2010.
- [5] C. Kaner, J. Falk, and H.Q. Nguyen, Testing Computer Software, 2nd ed., John Wiley & Sons, 1999.
- [6] D.J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, 2010.
- [7] R. Todnem By, “Organisational Change Management: A Critical Review,” J. Change Management, vol. 5, no. 4, 2005, pp. 369–380.
- [8] G.G. Claps, R. Berntsson Svensson, and A. Aurum, “On the Journey to Continuous Deployment: Technical and Social Challenges along the Way,” Information and Software Technology, vol. 57, 2015, pp. 21–31.
- [9] 2. H.H. Olsson, H. Alahyari, and J. Bosch, “Climbing the ‘Stairway to Heaven’—a Multiple Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software,” Proc. 38th EUROMICRO Conf. Software Eng. and Advanced Applications (SEAA 12), 2012, pp. 392–399.
- [10] M. Mäntylä et al., “On Rapid Releases and Software Testing: A Case Study and a Semisystematic Literature Review,” Empirical Software Eng., Oct. 2014, pp. 1–42

AUTHORS

Ning Luo is the senior software architect at Intel. His research interests include software requirements and architecture, continuous delivery, DevOps, and software product lines. Please contact him at ning.luo@intel.com.



Yue Xiong is the senior director of Intel. Please contact him at andy.xiong@intel.com.



MEASUREMENT OF SOFTWARE DEVELOPMENT EFFORT ESTIMATION BIAS: AVOIDING BIASED MEASURES OF ESTIMATION BIAS

Magne Jørgensen

Simula Metropolitan Center for Digital Engineering, Oslo, Norway

ABSTRACT

In this paper, we propose improvements in how estimation bias, e.g., the tendency towards under-estimating the effort, is measured. The proposed approach emphasizes the need to know what the estimates are meant to represent, i.e., the type of estimate we evaluate and the need for a match between the type of estimate given and the bias measure used. We show that even perfect estimates of the mean effort will not lead to an expectation of zero estimation bias when applying the frequently used bias measure: $(\text{actual effort} - \text{estimated effort})/\text{actual effort}$. This measure will instead reward under-estimates of the mean effort. We also provide examples of bias measures that match estimates of the mean and the median effort, and argue that there are, in general, no practical bias measures for estimates of the most likely effort. The paper concludes with implications for the evaluation of bias of software development effort estimates.

KEYWORDS

Software development effort estimation, measurement of estimation overrun, proper measurement of bias.

1. INTRODUCTION

Imagine that you are asked to throw two dice many times. Each time, the outcomes of the two dice are multiplied. The dice are fair, which gives 36 equally likely outcomes ranging from 1 (1x1) to 36 (6x6), with a mean outcome of 12.25 (3.5x3.5), a median outcome of 10 (half of the outcomes have values of 10 or less) and a mode of 6 (most frequently occurring value¹). What would be the best estimates of the products when the goal is to give unbiased estimates?

A good response would be that the best estimates depend on how the bias will be measured, i.e., how we will interpret unbiased estimates. We may, for example, interpret estimation bias as mean bias (a tendency to give estimates higher or lower than the mean), median bias (a tendency to give estimates higher or lower than the median), or even mode bias (a tendency to give estimates higher or lower than most frequently occurring value). Assume that you get informed that the goal is to give mean unbiased estimates, and consequently select the mean outcome (12.25) for each of your estimates. You are likely to have the correct prediction less often compared to the use of the mode (6) as your estimate and more than half of your estimates will be too high, but the mean outcome will be expected to be the same as your estimate, i.e., the bias measure:

¹ The value 12 is occurring equally often, but for simplicity, we choose 6 as our mode value in this example.

$$\frac{1}{N} \sum_{i=1}^N (act_i - est_i) = \frac{1}{N} \sum_{i=1}^N (act_i) - \frac{1}{N} \sum_{i=1}^N (est_i)$$

Has an expected value of zero, where act_i and est_i are the actual and estimated product of throw i , respectively.

What if you had used the relative (scale-free) measure:

$$\frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{act_i}$$

as your bias evaluation measure. In this case, as we will show later, neither the mean nor the median gives an expected bias of zero. Instead, this measure rewards under-estimating the mean a lot. The optimal value for this bias measure is in our example is 6, which happens to be the mode (but will not in general be the mode). If we, however, change the bias measure to:

$$\frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{est_i}$$

i.e., divide by the estimated rather than the actual values, the mean again becomes the estimate leading to an expectation of zero bias. The use of the median outcome as your estimate is for none of the three bias measures leading to an expectation of zero bias.

The above example is meant to demonstrate that without a match between type of estimates and how we evaluate the bias of the estimates, we cannot ensure evaluation fairness (giving perfect scores to perfect estimates of the intended type) and we risk rewarding poorer estimation performance. Implicitly, the example also tells us that without knowing what type of estimate we evaluate, it is hard to know how to evaluate it properly.

The example can easily be transformed into one of effort estimation. It illustrates, for example, that even perfect estimates of the mean effort will not give an expected estimation bias of zero when using the measure:

$$\frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{act_i}$$

and that observing an estimation bias of zero using this measure suggests that the estimates had a tendency to under-estimate the mean effort, not that the estimates are unbiased. Not knowing what type of effort estimate we evaluate makes it hard to evaluate their bias, e.g., degree of under-estimating the actual effort, properly.

The next section aims at giving more insight into the match and lack of match between the type of estimates and frequently used bias measures. While there has been much research on challenges with the use of common software development estimation *error* measures, in particular the challenge connected with the use of the Mean Magnitude of Relative Error (MMRE), see for example [1-5], we have been unable find analyses of challenges and proper use of software development effort estimation *bias* measures.

2. MEASURES OF THE BIAS OF EFFORT ESTIMATES

We propose that to enable fair evaluation of the bias, such as the degree of overrun, of effort estimates we need to:

- Identify the type of estimates to be evaluated, e.g., whether the estimates are intended to be estimates of the mean, median or mode (most likely) use of effort.
- Select a bias evaluation measure that matches the identified type of estimate. A match is in this context understood as that the measure gives zero bias for perfect estimates of the identified type of estimate. This criterion corresponds to what is the requirement for a proper scoring rule, see for example [6].

These conditions for meaningful evaluation of estimation bias are similar to those suggested for evaluation of estimation error, in [7, 8], i.e., the same conditions applied on bias instead of error. Notice that these requirements are not meant to be sufficient, just necessary conditions, for meaningful evaluation of estimation bias.

For the first step, identifying the type of estimate, we believe it is useful, perhaps even necessary, to apply a probabilistic view on effort usage, i.e., that the use of effort is uncertain and that the actual effort is one sample (draw) from a (typically unknown) probability effort distribution. Common types of estimates, with probabilistic interpretations, are estimates of the mean, the median, and the most likely (mode) effort. Estimates of the mean effort is, for example, what is intended produced by linear regression estimation models derived by using OLS (Optimized Linear Square), the median effort is frequently used as the planned effort or as input to the budget in several large-scale projects [9], and the most likely effort may be what the software developers give when they are asked about software task estimate [10]. The widely used PERT-model, see [11], is a good example of the use of different types of estimates with probabilistic interpretations in software development effort estimation contexts. Here the developers are asked to give the mode² (the most likely effort), together with the minimum and the maximum effort (or the 10 and 90% percentiles). The mean effort is then calculated based on these three values, typically using the formula:

$$\text{mean effort} = \frac{\text{min. effort} + 4 \text{ most likely effort} + \text{max. effort}}{6}$$

and used in the planning of the software projects [11].

In the following, we briefly discuss a selection of non-matching and matching bias measures for each of the three types of estimates, i.e., estimates of the mean, median and mode.

2.1. Assessment of bias of estimates of the mean effort

We start by showing, as claimed in the introduction, that the commonly used estimation bias (effort overrun) measure:

$$\text{mean RE}_{act} = \frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{act_i}$$

rewards underestimates of the mean efforts.

²The reason for not requesting the mean estimate, but instead the mode, may be that it is believed to be easier to give qualified expert judgments on the effort typically needed for the type of task (the most likely or mode effort), rather than the (perhaps more abstract in terms of looking back on prior experience) estimate of the mean use of effort.

Assume that we use the true mean effort as our estimate, i.e., we set the estimated effort (est_i) equal to the mean value (μ_i) of the software development effort distributions $I = 1 \dots N$. In other words, we have perfect estimates of the mean efforts. We then have that the expected value of this measure, due to the linearity of the mean, can be expressed as:

$$E \left[\frac{1}{N} \sum_{i=1}^N \frac{(act_i - \mu_i)}{act_i} \right] = \frac{1}{N} \cdot \left[E \left(\frac{act_1 - \mu_1}{act_1} \right) + \dots + E \left(\frac{act_N - \mu_N}{act_N} \right) \right] =$$

$$\frac{1}{N} \cdot \left[E \left(1 - \frac{\mu_1}{act_1} \right) + \dots + E \left(1 - \frac{\mu_N}{act_N} \right) \right] = 1 - \frac{1}{N} \cdot \left[E \left(\frac{\mu_1}{act_1} \right) + \dots + E \left(\frac{\mu_N}{act_N} \right) \right].$$

An approximation of the expected value of a ratio of stochastic variables, see for example [12], is:

$$E \left(\frac{X}{Y} \right) \approx \frac{\mu_X}{\mu_Y} - \frac{Cov(X, Y)}{\mu_Y^2} + \frac{Var(Y)\mu_X}{\mu_Y^3}$$

Replacing X with the mean (μ_i), which is our estimate of the i -th task, and Y with the random variable act_i (which by definition has as its expected (mean) value μ_i) give that, for all tasks i :

$$\left(1 - E \left(\frac{\mu_i}{act_i} \right) \right) \approx 1 - \frac{\mu_i}{\mu_i} - \frac{Cov(\mu_i, act_i)}{\mu_i^2} + \frac{Var(act_i)\mu_i}{\mu_i^3} = \frac{Cov(\mu_i, act_i)}{\mu_i^2} + \frac{Var(act_i)}{\mu_i^2}$$

We have that $Cov(\mu_i, act_i) = 0$, since the correlation between the mean (μ_i) and the sampled values (act_i) will be zero. This implies that we should, for a perfect estimate of the mean effort, expect a bias towards over-estimation of effort of size:

$$\frac{Var(act_i)}{\mu_i^2}$$

when using the mean RE_{act} as our measure of bias, i.e., the bias of the bias measure (when evaluating estimates of the mean effort) increases with the variance of the actual effort.

Interestingly, if we use the slightly modified measure:

$$mean RE_{est} = \frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{est_i}$$

i.e., when we divide by the estimated rather than the actual effort, as implemented in for example [13], we have an expected bias of zero when the mean efforts are used as the estimates:

$$E \left[\frac{1}{N} \sum_{i=1}^N \frac{(act_i - \mu_i)}{\mu_i} \right] = \frac{1}{N} \cdot \left[E \left(\frac{act_1 - \mu_1}{\mu_1} \right) + \dots + E \left(\frac{act_N - \mu_N}{\mu_N} \right) \right]$$

$$= \frac{1}{N} \cdot \left[E(act_1 - \mu_1) \cdot E \left(\frac{1}{\mu_1} \right) + \dots + E(act_N - \mu_N) \cdot E \left(\frac{1}{\mu_N} \right) \right]$$

$$= \frac{1}{N} \cdot \left[(E(\mu_1) - E(\mu_1)) \cdot E \left(\frac{1}{\mu_1} \right) + \dots + (E(\mu_N) - E(\mu_N)) \cdot E \left(\frac{1}{\mu_N} \right) \right] = 0$$

due to independence between $(act_i - \mu_i)$ and μ_i .

Perfect estimates of the mean effort also, due to the linearity of the mean values and as exemplified in the introduction, give zero bias for:

$$\text{mean } RE_{dev} = \frac{1}{N} \sum_{i=1}^N (act_i - est_i).$$

Our results consequently shows that we should use:

$$\frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{est_i} \text{ or } \frac{1}{N} \sum_{i=1}^N (act_i - est_i)$$

as our estimation bias measure if we want to give zero bias to perfect estimates of the mean. It also shows that we should stop using

$$\frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{act_i}$$

as a bias measure when evaluating estimates of the mean. Using this bias measure, we will not only evaluate perfect estimates of the mean as biased towards over-estimation, but also reward under-estimation of the mean effort. For example, a seemingly unbiased estimation model, giving a mean RE_{est} of zero, has in reality given estimates with a bias towards under-estimating the mean effort with the value of:

$$\frac{Var(act_i)}{\mu_i^2}$$

per estimate.

2.2. Assessment of bias of estimates of the median effort

Measurement of the bias of estimates of the median effort, with the requirement of zero bias for perfect estimates of the median, is trivial for bias measures based on the median, rather than the mean, deviation.

In such cases, we have that the median of all three measures:

$$(act_i - est_i), \frac{(act_i - est_i)}{act_i} \text{ or } \frac{(act_i - est_i)}{est_i}, i = 1..N,$$

results in zero bias for perfect estimates of the median effort. This is the case since the median, by definition, is the value that is just as likely to exceed as not to exceed, i.e., half of the observations will have positive and the other half negative deviations between the actual and the estimated values. Dividing this deviation by the estimated or the actual value does not affect this relationship. All the above measures are consequently proper measures of median bias of effort estimates.

Another median-matching bias measure is based on the logarithm of the ratio of the estimated and the actual effort (log-error). This measure, proposed in amongst others [14], has the advantage that it combines being a relative (scale-free) measure with symmetric bias values. As is easy to see, the relative bias measures:

$$\frac{(act_i - est_i)}{act_i} \text{ and } \frac{(act_i - est_i)}{est_i}$$

are *not* symmetric around 0, i.e., the possible scores are in the interval $(-\infty, 1)$ and $(-1, \infty)$, respectively, which for large estimation errors give much higher penalties for deviations in one direction compared to the other.

The proposed, symmetric and median unbiased, measure, which we term MdLogErr, is defined as the:

$$\text{median of } \log\left(\frac{act_i}{est_i}\right) = \text{median of } \log(act_i) - \log(est_i), \text{ for } i = 1 \dots N,$$

The expected value of MdLogErr is zero for perfect estimates of the median due to the preservation of the percentiles, including the median (which is the 50% percentile) when back-transforming the log-transformed values, i.e., if 50% of $\log(act_i)$ is above $\log(median_i)$, then 50% of the act_i will also be above the $median_i$. Use of the log-error may be said to have non-intuitive interpretation of bias, given that it is based on log-scores rather than percentage deviation. The interpretation of the scores of asymmetric bias measures is, however, not trivial either, given the different penalties for high over- and under-run. A possible selection criterion is to use asymmetric measures like

$$\frac{(act_i - est_i)}{est_i}$$

when there are no large deviations between the estimated and actual effort expected, in which case the asymmetry would not complicate the interpretations, and to select

$$\log\left(\frac{act_i}{est_i}\right)$$

otherwise.

2.3. Assessment of bias of estimates of the mode effort

The assessment of bias of estimates of the mode (most likely) effort introduces several evaluation challenges. If we know the percentiles of the modes, we may (similarly to how we evaluate the median) measure mode bias through measures of calibration and informativeness (or sharpness), see for example [7, 15]. For example, if the mode values, on average, were at the 45% percentile, then unbiased estimates of the mode should be higher than exactly 45% of the actual effort values. Unfortunately, we typically do not know the percentiles of the mode values.

As pointed out in several studies, see for example [16, 17], there are inherent problems in evaluating predictions of the mode. To our knowledge, there are currently no practical measures enabling bias evaluations of estimates of the mode, other than through the hit rate of its percentiles.

3. IMPLICATIONS AND CONCLUSION

Possibly, the great majority of previous studies on software development effort estimation bias may be accused of doing what Gneiting [6] describes as: “*The common practice of requesting*

‘some’ point forecast, and then evaluating the forecasters by using ‘some’ (set of) scoring function(s), is not a meaningful endeavor.” As demonstrated in this paper, meaningful evaluation of estimation bias requires both that we know what we evaluate and that we select bias evaluation measures that match the type of estimates we evaluate, i.e., that we have a proper scoring rule. Otherwise, we may produce misleading results and/or give incentives for strategically too low or too high estimates. If, for example, an estimator knows that the estimates will be evaluated by its average percentage overrun, i.e., the measure:

$$\text{mean } RE_{act} = \frac{1}{N} \sum_{i=1}^N \frac{(act_i - est_i)}{act_i}$$

the estimator will benefit from giving estimates lower than the mean effort instead of giving honest estimates of the mean. The higher the variance in the use of effort per project or task, the lower are the estimates that give an expectation of zero average effort overrun using $\text{mean } RE_{act}$.

As an illustration of the possible implications of different types of estimates on measures of bias, assume that the actual use of effort of a hypothetical software project is distributed as in Figure 1. The distribution in Figure 1 is a log-normal distribution³ with mean of 236 work-hours, a median of 209 work-hours, a mode (most likely effort) of 162 work-hours, and a standard deviation of 126 work-hours. We calculated the expected estimation bias by simulating that the project is executed 10.000 times and that the actual effort is for each project execution randomly drawn from the distribution in Figure 1, i.e., we assume a hypothetical repeated execution of the same project without learning in the same context. We then calculated the expected (mean) RE_{act} in the case of that we use perfect estimates⁴ of the mode, the median or the mean of the actual effort distribution as our effort estimate. From this simulation we found that if the estimator gave the most likely effort (162 work-hours) as his/her project effort estimate, the expected RE_{act} would be 12% (12% effort overrun), while the use of the median effort (209 work-hours) as the estimate would give an expected RE_{act} of -14% (14% effort under run), and use of the mean effort (236 work-hours) as the estimate would give an expected RE_{act} of -28% (28% effort under run). The 28% effort under run is very close to what is expected from the formula we derived earlier, i.e., the expected effort under run when estimating the mean effort and using the RE_{act} as our measure of estimation bias is:

$$\frac{Var(act_i)}{\mu_i^2} = \frac{126^2}{236^2} = 0.29$$

The estimate that gives an expected RE_{act} of zero is in this case approximately 185 work-hours, which is neither the mode, the median or the mean of the distribution of actual effort.

Our illustration documents that the measured bias varies much dependent on what type of estimate the developer gives. Also, the illustration shows that it will be hard to know what type of estimate to give, to have an expectation of zero bias, when using the bias measure RE_{act} . Giving

³As argued in, for example [10], a log-normal distribution reflects typical properties of effort usage distributions, e.g., that it is right-skewed and has a minimum of zero.

⁴Clearly, in realistic situations we do not know with high accuracy the distribution of actual effort of a software project, but have to estimate this, as well. For illustrative purposes on properties of the bias measure, however, the assumption of perfect knowledge about the underlying effort distribution (effort uncertainty) is considered useful. Similarly, while we will never repeat the same project without learning in the same context, this assumption is useful to illustrate the properties of the bias measures. The core result do not change if we, for example, replace the actual mean of the effort distribution (assuming a perfect estimator) with the estimated mean (assuming that the estimator just have an estimate of the underlying mean use of effort), or that we replace the assumption of repeating the same project without learning with executing different projects and learning.

perfect estimates of the mean effort of a project will, for example, always⁵ result in an expected RE_{act} larger than zero (effort under run).

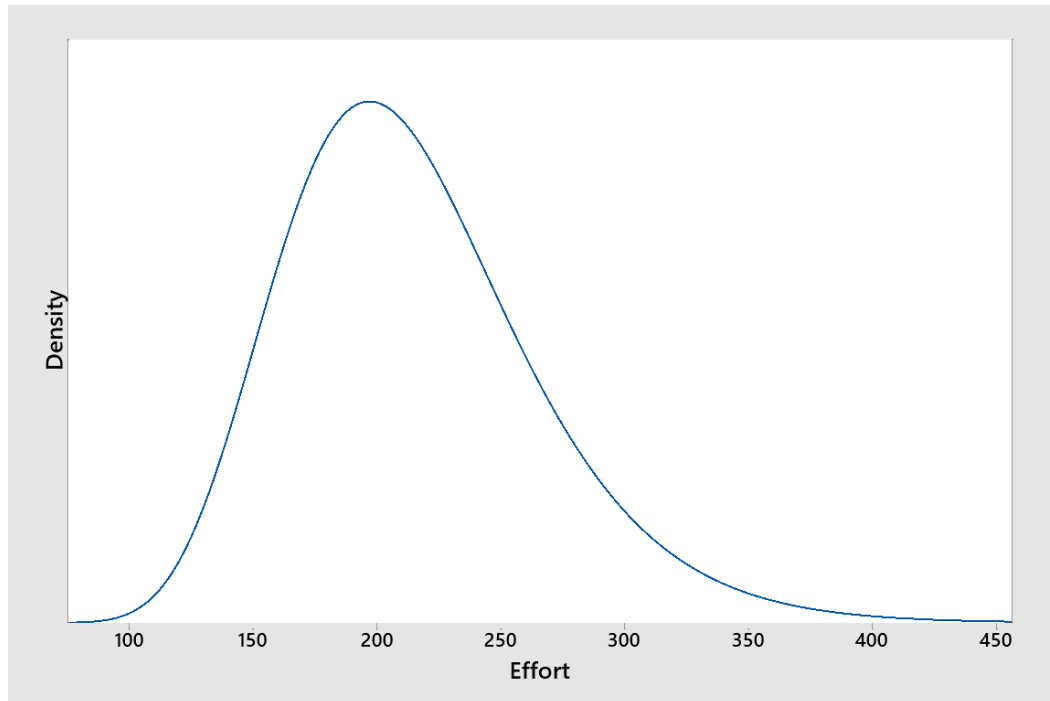


Figure 1. Hypothetical distribution of effort usage

In many estimation bias evaluation contexts, the type of estimates to be evaluated will not be explicitly described, but instead simply be presented as effort ‘estimates’. In those contexts, to enable fair and meaningful bias evaluations, we will have to make qualified guesses about what the estimates are meant to represent. Such qualified guesses may be made derived from the optimization function used to derive the model (e.g., OLS-regression, PERT-formula, etc.), the purpose of the estimate (e.g., input to budget or plan), or the estimation instructions (e.g., requesting the “most likely” use of effort). If the estimates to be evaluated are believed to be of various types we may either evaluate different types of estimates separately or evaluate the estimates based on the dominant type of estimate.

To continue as before, with little or no reflection on the type of estimate or concerns about a match between the type of estimate and evaluation measure, means that we will continue having a hard time interpreting how biased the effort estimates are and how large the cost overruns in software development really are.

REFERENCES

- [1] J. S. Armstrong, "A commentary on error measures," *International Journal of Forecasting*, vol. 8, pp. 99-111, 1992.
- [2] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, "A simulation study of the model evaluation criterion MMRE," *Ieee Transactions on Software Engineering*, vol. 29, no. 11, pp. 985-995, 2003.
- [3] B. A. Kitchenham, L. M. Pickard, S. G. MacDonell, and M. J. Shepperd, "What accuracy statistics really measure," *IEE Proceedings-Software*, vol. 148, no. 3, pp. 81-85, 2001.

⁵It is only when there is no variance in the use of effort that perfect estimates of the mean will give zero bias. Clearly, no variance in use of effort is unrealistic in the context of software development projects.

- [4] I. Myrtveit and E. Stensrud, "A controlled experiment to assess the benefits of estimating with analogy and regression models," *Ieee Transactions on Software Engineering*, vol. 25, no. 4, pp. 510-525, 1999.
- [5] M. Jørgensen, "A critique of how we measure and interpret the accuracy of software development effort estimation," in *First international workshop on software productivity analysis and cost estimation*, 2007: Citeseer.
- [6] T. Gneiting, "Making and evaluating point forecasts," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 746-762, 2011.
- [7] M. Jørgensen, "Evaluating probabilistic software development effort estimates: Maximizing informativeness subject to calibration," *Information and software Technology*, vol. 115, pp. 93-96, 2019.
- [8] M. Jørgensen, M. Welde, and T. Halkjelsvik, "Evaluation of Probabilistic Project Cost Estimates," *IEEE Transactions on Engineering Management*, vol. In press, 2021.
- [9] K. F. Samset, G. H. Volden, N. Olsson, and E. V. Kvalheim, "Governance schemes for major public investment projects: A comparative study of principles and practices in six countries," ed: Ex ante akademisk forlag, 2016.
- [10] T. Halkjelsvik and M. Jørgensen, *Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life*. Springer, 2018.
- [11] D. Golenko-Ginzburg, "On the distribution of activity time in PERT," *Journal of the Operational Research Society*, vol. 39, no. 8, pp. 767-771, 1988.
- [12] A. Stuart, S. Arnold, J. K. Ord, A. O'Hagan, and J. Forster, *Kendall's advanced theory of statistics*. Wiley, 1994.
- [13] K.-S. Na, J. T. Simpson, X. Li, T. Singh, and K.-Y. Kim, "Software development risk and project performance measurement: Evidence in Korea," *Journal of Systems and Software*, vol. 80, no. 4, pp. 596-605, 2007.
- [14] L. Törnqvist, P. Vartia, and Y. O. Vartia, "How should relative changes be measured?," *The American Statistician*, vol. 39, no. 1, pp. 43-46, 1985.
- [15] T. Gneiting and M. Katzfuss, "Probabilistic forecasting," *Annual Review of Statistics and Its Application*, vol. 1, pp. 125-151, 2014.
- [16] T. Gneiting, "When is the mode functional the Bayes classifier?," *Stat*, vol. 6, no. 1, pp. 204-206, 2017.
- [17] C. Heinrich, "The mode functional is not elicitable," *Biometrika*, vol. 101, no. 1, pp. 245-251, 2014.

AUTHOR

Magne Jørgensen received a Dr. Scient degree in informatics from University of Oslo in 1994. He has 10 years of industry experience as software developer, project leader and manager and worked for 20 years as a professor at University of Oslo. He is currently a chief research scientist at Simula Metropolitan and a professor at Oslo Metropolitan University. He is one of the founders of evidence-based software engineering. Current research interests include software management, psychology of human judgement and cost estimation. (H-index 52).



BUILD AUTOMATION TOOLS FOR SOFTWARE DEVELOPMENT

A COMPARATIVE STUDY BETWEEN MAVEN, GRADLE AND BAZEL

Mridula Prakash

L&T Technology Services, CTO Office, Mysore, India

ABSTRACT

The automated processes will play an increasingly vital role in continuous integration as the pace of design and development of new software picks up. With the importance of software build automation tools taking center stage, the present paper undertakes a comparative analysis of three best available solutions - Maven, Gradle and Bazel. We aim to evaluate the solutions based on their efficiency and performance in the context of software build automation and deployment. There are some fundamental differences in the way each tools approach builds. The aim of this study is also to provide the reader with a complete overview of the selected build automation tools and, the relevant features and capabilities of interest. In addition, the paper leads to a broader view on the future of the build automation tools ecosystem.

KEYWORDS

Automated process, Build automation tools, Maven, Gradle, Bazel.

1. INTRODUCTION

The build automation process involves automating tasks about software build, including the compilation of source code into binary code, packaging the binary code, and running the automated tests; as the final procedure. The process helps in reducing down time, optimizing costs, and simplifying the overall development process. Over the years, given the growing demand for software development globally, a series of tools have emerged to provide a streamlined continuous integration framework.

Build automation tools can be broadly classified into two types namely:

1. Build automation utility
2. Build automation servers

The build automation utilities consist of a range of solutions, including, Gradle, Maven, Bazel, and Cmake. Their primary purpose of these tools is to generate software builds by compiling and linking the source code. The build automation server, on the other hand, comprises of continuous integration-based web servers, with instances including continuous management tools, and continuous integration tools. We first provide a comprehensive overview of the emerging tools like Gradle, Maven and Bazel and then proceed to summarize with the latest trends and the way forward.

1.1. How to Select the Best Build Automation Tool?

An organization needs to select an appropriate tool because it will save a lot of money and time along with providing the best quality outcome. Before selecting an automation tool everyone has to go through the following simple steps,

- Step 1: Collect all the requirements.
- Step 2: Categorize your requirements (basic, technical, business, feature).
- Step 3: Create a list of tools against basic requirements.
- Step 4: Shortlist tools against other requirements.
- Step 5: Create a chart for comparison.
- Step 6: Scorecard.

The initial step is to collect all the requirements that are needed for our project. The next step is to categorize the requirements i.e split the requirements as basic, technical, feature and business.

- Basic Requirements deal with the basic needs of the project like what type of product or application we are going to build and the ease of execution.
- Technical Requirements consists of system requirements along with technical details like microprocessor details, configuration, the platform used and operating system specifics.
- Business Requirements concentrate on the cost, budget, time and deadline of the project.
- Features Requirements are all about what kind of feature we are looking for like alarm descriptions, feature from user perspective, user interface details, integration and reporting.

After categorizing the requirements, the tools must be mapped with all types of requirements to make sure whether the tool is satisfying all the needs.

At last, a comparison chart has to be plotted using all the details and generate the scorecard which will let us know which tool is best suited for our project.

Open source and multi-language, multi-module supported tools like Gradle, Maven and Bazel are becoming very popular in continuous integration. In this paper, we present an in-depth view into the framework and the specification of each of the tools to understand their feasibility and applicability across projects.

2. MAVEN

Maven is an open-source Java-based build automation tool, used mostly for the creation of Java projects. The tool was created by Jason Van Zyl in 2002, with its first release on July 13, 2004 by the Apache Software Foundation.

The tool can also be used to build and manage projects in other languages such as Ruby, Scala and C# respectively. The working mechanism of Maven consists of usage of declarative speech approach, where the project structure and the contents are described in the form of the project object model.

Maven attempts to apply patterns to any project's build structure. It promoted comprehension and productivity by making use of the best practices. It acts as a management tool in managing the documentation, builds and dependencies.

2.1. Project Creation in Maven

STEP-1: Download the Maven software from the official website:

<https://maven.apache.org/download.cgi>

STEP-2: Install the software by changing the appropriate environment variables.

STEP-3: Set path in the root directory or directory of your choice.

STEP-4: Check for successful installation using the command `maven -v`.

2.2. Maven Build Phases

Maven build lifecycle goes through a series of stages, called as default build phases – validate, compile, test, package, verify, install and deploy. These phases are executed sequentially when we run a maven build command. Let us take an example to understand Maven in detail.

Build a java project using maven

Maven supports two ways to build the project as below:

1. Using eclipse
2. Using command prompt

Using Eclipse

Create a Maven project and then check for the POM file which is very important for automation. In the POM file, we add the dependencies. As mentioned in Figure 1, we can take the dependencies from the system which we are using or we can use online repositories. Usually java projects use the test frames like junit or testing. We can use any test frame of our choice.

Example dependency used in my Pom file.

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>7.4.0</version>
  <scope>test</scope>
</dependency>
```

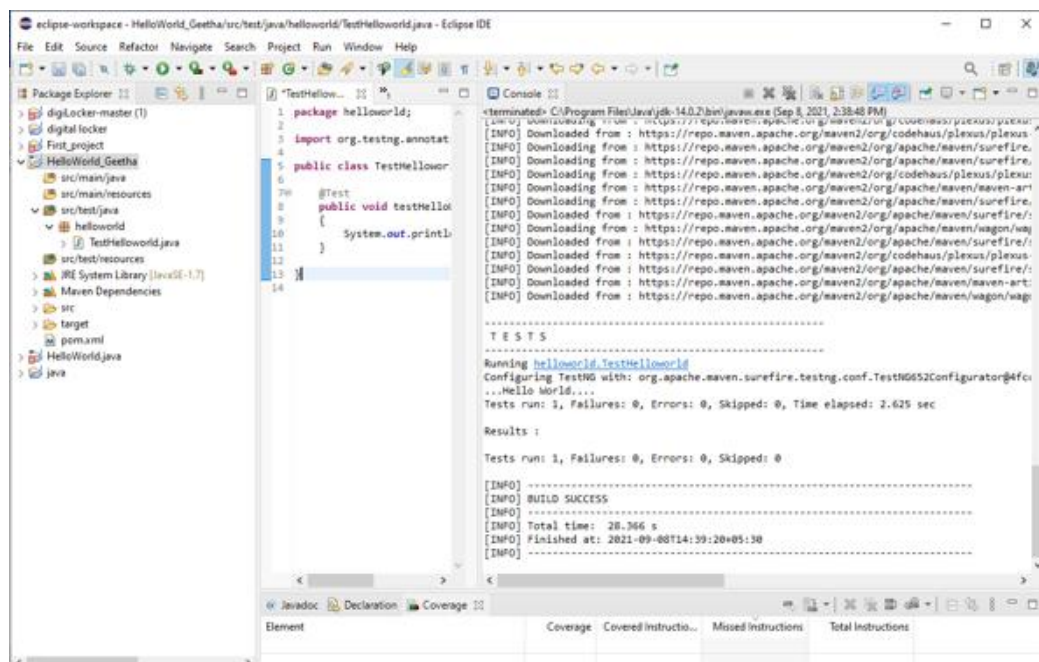



Figure 3: Project build compilation in Eclipse using Maven

To verify we can check using the command prompt. Go to the folder in which our project is saved and then use the maven test command.

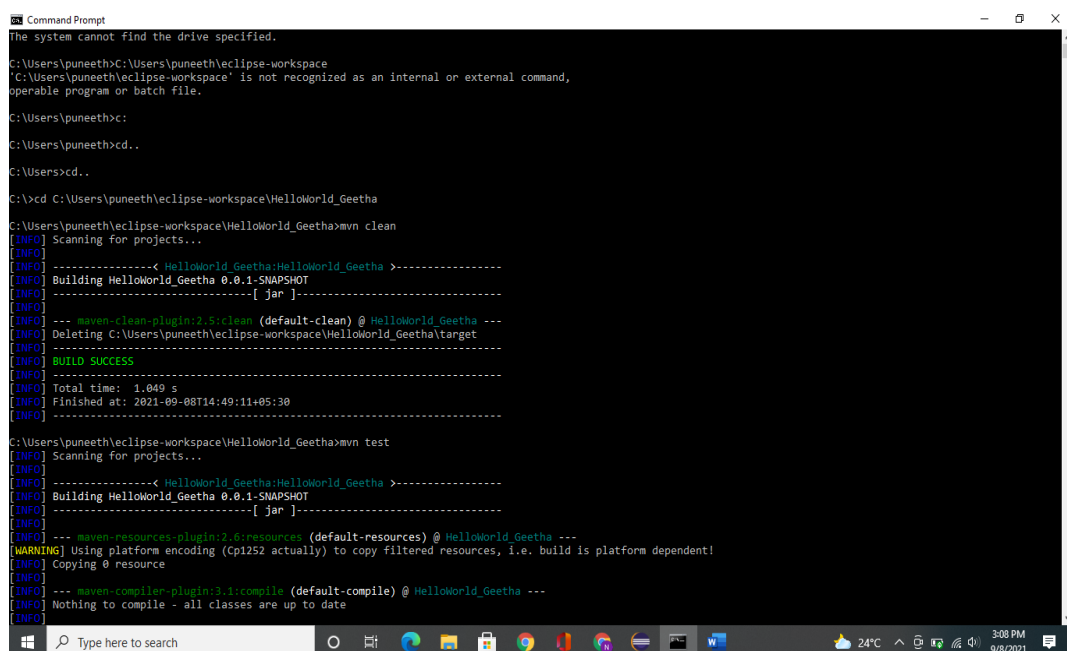


Figure 4: Project build compilation in command prompt using Maven

3. GRADLE

Gradle began as an open-source project led by Hans Dockner and Adam Murdoch. The initial success of the Gradle build tool project paved the way for the establishment of the Gradle Enterprise in 2017.

As a build automation tool, Gradle supports multiple-language software development. The tool controls various processes involved in the software development processes. It automates processes such as the tasks of compilation, packaging, testing, deployment and publishing as well. The languages supported by the Gradle build automation tool, include:

- Java
- Kotlin
- Groovy
- Scala
- C/C++, and
- JavaScript

The working mechanism of the Gradle tool consists of inclusion of directed acyclic graph to detect the order in which each of the tasks are running. The tool operates by leveraging certain series of build tasks which can run either serially or in parallelly. An important part of any build tool is the ability to avoid doing work that has already been done. It supports incremental builds by reading the updated parts of the build tree. You need to tell Gradle which task properties are inputs and which are outputs. If a task property affects the output, be sure to register it as an input, otherwise the task will be considered up to date when it's not. Gradle supports cache conflicts and dependency management of the builds. It also supports multi-project organization.

3.1. Project Creation in Gradle

STEP-1: Download the Gradle software from the official website - <https://gradle.org/install/>

STEP-2: Install the software by changing the appropriate environment variables.

STEP-3: Set path in the root directory or directory of your choice.

STEP-4: Check for successful installation using the command `gradle -version`.

STEP-5: Once details of Gradle version is displayed, create a build using the `gradle init` command.

STEP-6: On pressing enter, select the language of your choice for project creation.

STEP-7: Project starts generated corresponding to the program in the build. Gradle file.

STEP-8: Project is successfully created in Gradle.

The building process includes compiling, linking and packaging the code. The tool is supported mostly for a groovy-based domain. This tool provides building, testing and deploying of the software. It is used to build any software and large projects. Gradle mainly focuses on maintenance, performance and flexibility.

3.2. Gradle Build Phases

A Gradle build has three distinct phases.

- 1 **Initialization:** Gradle supports single and multi-project builds. During the initialization phase, Gradle determines which projects are going to take part in the build, and creates a Project instance for each of these projects.

- 2 **Configuration:** During this phase, the project objects are configured. The build scripts of all projects which are part of the build are executed.
- 3 **Execution:** Gradle determines the subset of the tasks, created and configured during the configuration phase, to be executed. The subset is determined by the task name arguments passed to the Gradle command and the current directory. Gradle then executes each of the selected tasks.

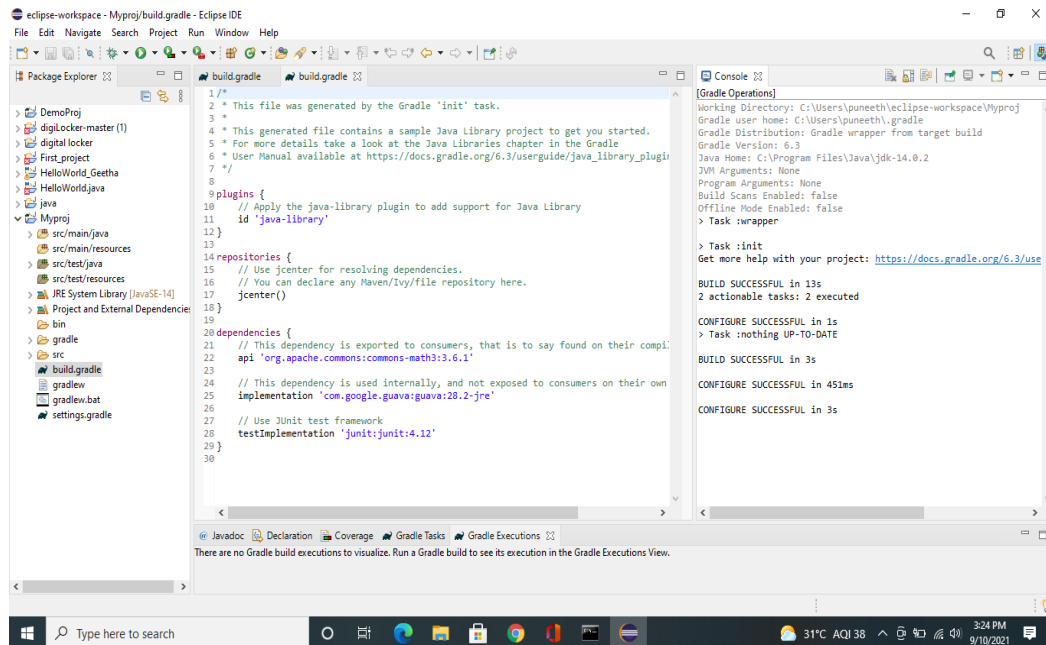


Figure 5: Project build compilation in eclipse using Gradle

Gradle project creation is performed using eclipse and the results are as shown in the above picture. The process to build the application using Gradle is almost the same as Maven. The difference is the Maven Project will be built only after modifying the POM file, but in Gradle, the application is built just after the creation of the project.

Gradle has many options in the command prompt to select the project details and all other requirements to start the build process. The command used to start the build process in Gradle is “Gradle init”.

After entering this command in the command prompt, it will initialize the task and starts showing all the options one by one and we have to select what are the requirements needed to build the application.

```
Microsoft Windows [Version 10.0.19042.1202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\puneeth>gradle init
Starting a Gradle Daemon (subsequent builds will be faster)

Select type of project to generate:
 1: basic
 2: application
 3: library
 4: Gradle plugin
Enter selection (default: basic) [1..4] 2

Select implementation language:
 1: C++
 2: Groovy
 3: Java
 4: Kotlin
 5: Scala
 6: Swift
Enter selection (default: Java) [1..6] 3

Split functionality across multiple subprojects?:
 1: no - only one application project
 2: yes - application and library projects
Enter selection (default: no - only one application project) [1..2] 1

Select build script DSL:
 1: Groovy
 2: Kotlin
Enter selection (default: Groovy) [1..2]

Select test framework:
 1: JUnit 4
 2: TestNG
 3: Spock
 4: JUnit Jupiter
Enter selection (default: JUnit 4) [1..4] 4

Project name (default: puneeth): MyGradleProj
Source package (default: MyGradleProj): com.kkjavatutorials

> Task :init
Get more help with your project: https://docs.gradle.org/6.9.1/samples/sample_building_java_applications.html
```

Figure 6: Options display to build a process in Gradle tool.

```
4: Gradle plugin
Enter selection (default: basic) [1..4] 2

Select implementation language:
 1: C++
 2: Groovy
 3: Java
 4: Kotlin
 5: Scala
 6: Swift
Enter selection (default: Java) [1..6] 3

Split functionality across multiple subprojects?:
 1: no - only one application project
 2: yes - application and library projects
Enter selection (default: no - only one application project) [1..2] 1

Select build script DSL:
 1: Groovy
 2: Kotlin
Enter selection (default: Groovy) [1..2]

Select test framework:
 1: JUnit 4
 2: TestNG
 3: Spock
 4: JUnit Jupiter
Enter selection (default: JUnit 4) [1..4] 4

Project name (default: puneeth): MyGradleProj
Source package (default: MyGradleProj): com.kkjavatutorials

> Task :init
Get more help with your project: https://docs.gradle.org/6.9.1/samples/sample_building_java_applications.html

BUILD SUCCESSFUL in 4m 18s
2 actionable tasks: 2 executed
C:\Users\puneeth>
```

Figure 7: Complete testing of compilation process and build creation in Gradle tool.

```

C:\Users\puneeth> gradle init
2: yes - application and library projects
Enter selection (default: no - only one application project) [1..2] 1

Select build script DSL:
1: Groovy
2: Kotlin
Enter selection (default: Groovy) [1..2]

Select test framework:
1: JUnit 4
2: TestNG
3: Spock
4: JUnit Jupiter
Enter selection (default: JUnit 4) [1..4] 4

Project name (default: puneeth): MyGradleProj
Source package (default: MyGradleProj): com.kkjavatutorials

> Task :init
Get more help with your project: https://docs.gradle.org/6.9.1/samples/sample_building_java_applications.html
BUILD SUCCESSFUL in 4m 10s
2 actionable tasks: 2 executed
C:\Users\puneeth> gradle init

> Task :init SKIPPED
The settings file 'settings.gradle' already exists. Skipping build initialization.
BUILD SUCCESSFUL in 4s
C:\Users\puneeth> gradle init

> Task :init SKIPPED
The settings file 'settings.gradle' already exists. Skipping build initialization.
BUILD SUCCESSFUL in 2s
C:\Users\puneeth> gradle init

> Task :init SKIPPED
The settings file 'settings.gradle' already exists. Skipping build initialization.
BUILD SUCCESSFUL in 2s
C:\Users\puneeth>

```

Figure 8: Complete testing of compilation process and build creation in Gradle tool.

If observed the build compilation time in Gradle is more than Maven in the command prompt, this is due to the extra time consumed while opting for the requirements and automatic cache cleaning function of Gradle when a new application is started.

In Eclipse IDE, we can see that the build time of Gradle is very less compared to Maven.

4. BAZEL

Bazel is an open-source version of Google's internal solution, Blaze which is used for automating software processes within the search engine giant. The tool, an anagram of Blaze was launched in March 2015.

In terms of build automation tools, Bazel is similar to Apache Maven and Apache Ant. It provides complete automation in build and testing of the software, across platforms. The working mechanism consists of building of software applications from the source-code by following a certain a set of rules. These rules are created in Starlark Language which serves as dialect of the Python programming language.

The software application packages produced by Bazel capable of being integrated with Android and iOS platforms, thereby paving way for real-time applications. Built-in rules are present in Bazel for building software applications with various types of language support namely: C++, Java, Go, Python, Objective-C and Bourne Shell script respectively.

Bazel leverages parallelization techniques to speed up the process of build and software application package creation as well. New build rules can be written in Bazel so as enhance cross-platform integration. The tool also provides plugin-in support with the following IDEs: IntelliJ, Android Studio, and CLion.

4.1. Project Creation in Bazel

STEP-1: Download the Bazel software from the official

website.<https://docs.bazel.build/versions/main/install-windows.html>

STEP-2: Install the software by changing the appropriate environment variables.

STEP-3: Set path in the root directory or directory of your choice.

STEP-4: Check for successful installation using the command `bazel -v`.

STEP-5: Once details of Bazel version is displayed, create a build file with sub file called `planning.java`.

STEP-6: Create a sub-directory with a java file as `source.java`

STEP-7: Once build is created test the builds.

STEP-8: Project is successfully created in Bazel.

4.2. Bazel Build Phases

In Bazel, a build occurs in three distinct phases

- 1 **Loading Phase:** The first is loading during which all the necessary BUILD files for the initial targets, and their transitive closure of dependencies, are loaded, parsed, evaluated and cached. Errors reported during this phase include: package not found, target not found, lexical and grammatical errors in a build file, and evaluation errors.
- 2 **Analysis Phase:** The second phase, the analysis involves the semantic analysis and validation of each build rule, the construction of a build dependency graph, and the determination of exactly what work is to be done in each step of the build. Errors reported at this stage include: inappropriate dependencies, invalid inputs to a rule, and all rule-specific error messages.
- 3 **Execution Phase:** This phase ensures that the outputs of each step in the build are consistent with its inputs, re-running compilation/linking/etc. tools as necessary. Errors reported during this phase include: missing source files, errors in a tool executed by some build action, or failure of a tool to produce the expected set of outputs.

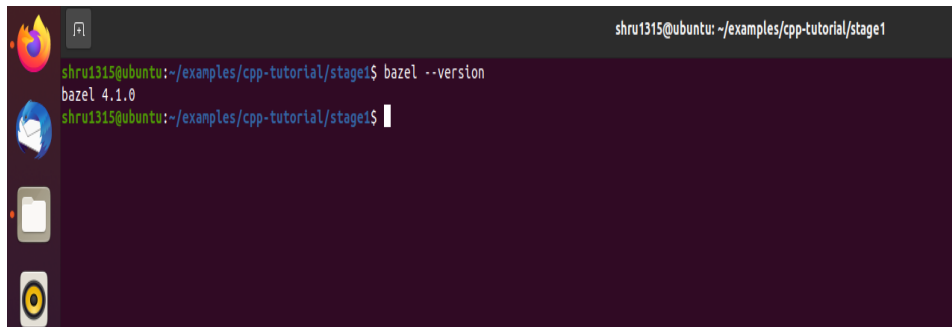
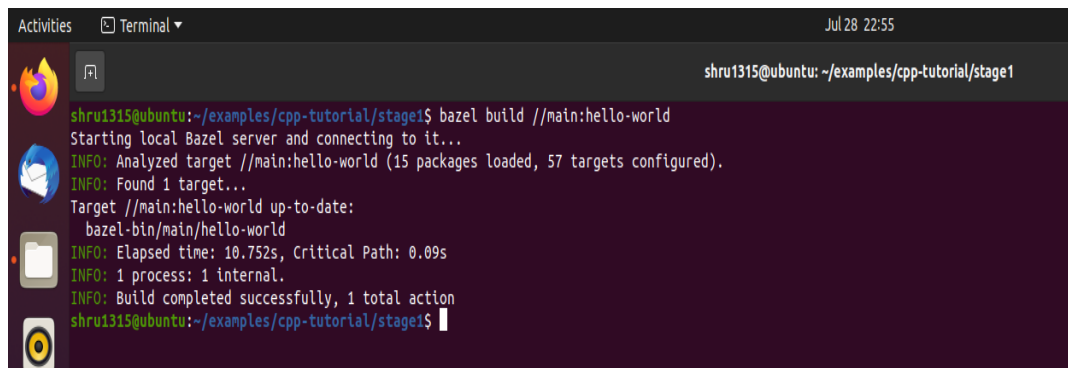
A screenshot of a Linux terminal window. The window title is 'shru1315@ubuntu: ~/examples/cpp-tutorial/stage1'. The terminal shows a command prompt 'shru1315@ubuntu:~/examples/cpp-tutorial/stage1\$' followed by the command 'bazel --version'. The output of the command is 'bazel 4.1.0'. The prompt is then followed by 'shru1315@ubuntu:~/examples/cpp-tutorial/stage1\$' with a cursor. The terminal has a dark background and a sidebar on the left with icons for Firefox, a file manager, and a terminal.

Figure 9: Step1: Installing the Bazel software.

A terminal window titled 'Terminal' with a date and time 'Jul 28 22:55' in the top right corner. The terminal shows a user 'shru1315@ubuntu' in the directory '~/examples/cpp-tutorial/stage1' running the command 'bazel build //main:hello-world'. The output shows the Bazel server starting, target analysis, and successful build completion.

```
shru1315@ubuntu: ~/examples/cpp-tutorial/stage1
shru1315@ubuntu:~/examples/cpp-tutorial/stage1$ bazel build //main:hello-world
Starting local Bazel server and connecting to it...
INFO: Analyzed target //main:hello-world (15 packages loaded, 57 targets configured).
INFO: Found 1 target...
Target //main:hello-world up-to-date:
  bazel-bin/main/hello-world
INFO: Elapsed time: 10.752s, Critical Path: 0.09s
INFO: 1 process: 1 internal.
INFO: Build completed successfully, 1 total action
shru1315@ubuntu:~/examples/cpp-tutorial/stage1$
```

Figure 10: Step 2: Checking for complete compilation process of Bazel tool.

5. CURRENT SCENARIO: PRACTICES, AND CHALLENGES

Build automation techniques and tools have emerged as a significant component for ensuring seamless and continuous integration. In the present-day scenario of build automation, there are certain practices which are being currently deployed to improve the build automation processes. Some of these include:

- Maintenance of a central code repository.
- Making the build as self-testing.
- Testing in a clone of the production management.
- Quicken up the build time execution.
- Maintenance of revision control system for the project's source code.

Although, build automation tools have several benefits, they have their own unique challenges, including:

- Selection of the right tool.
- Correspondence of project requirements and tool features.
- Scalability of the builds and its deployment.
- Expectations of the build to detect occurrence of new defects during the project creation.

The industry-wide adoption of our three selected tools is shown below [Figure 11]. The adoption of Maven has continued to rise steadily over the years, Bazel has been in decline around. There is a sharp uptick in the adoption of Gradle.

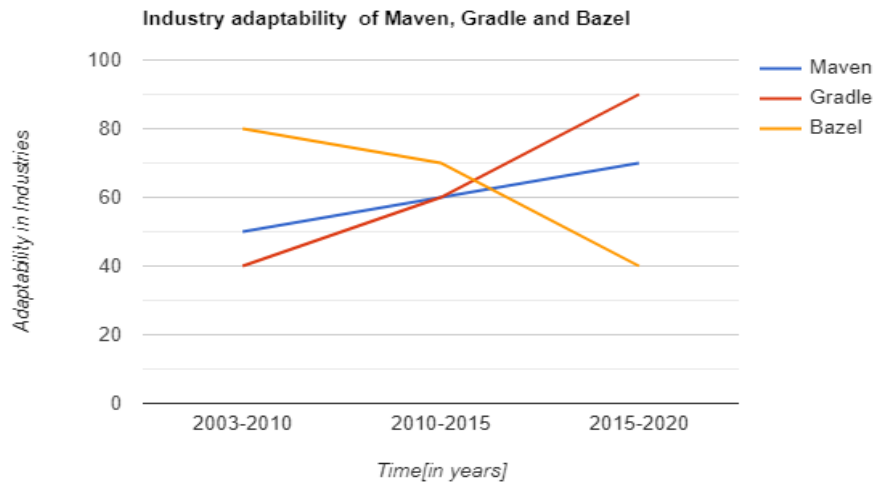


Figure 11: Line graph based on timeline analysis of tools

5.1. Maven Trends

1. Provides allowance to a developer to understand the entire state of a development within a short period of time.
2. Provision of easy guidance for project creation.
3. Maven provides more than 200+ plugin-in to work with which is one of the remarkable trends seen in the Maven tool.

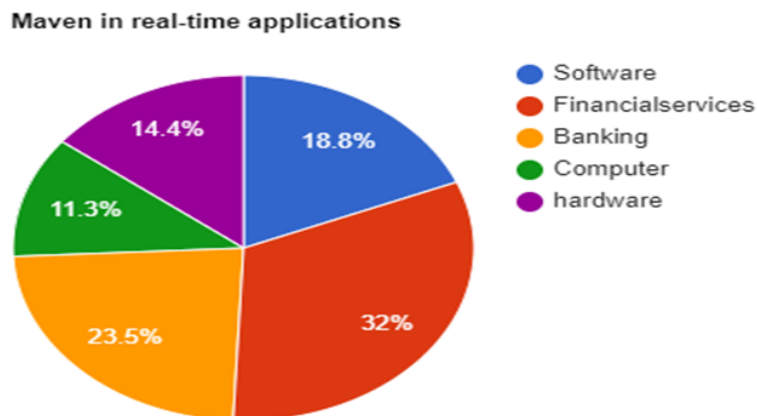


Figure 12: Maven real-time applications [Sample size: 100] (Source Material:From References*)

5.2. Maven Challenges

1. Tool is unreliable since it can collapse the software development lifecycle due to the limitless number of loops present in IDE.
2. Existence of pom.xml files make it difficult for usage when it comes to large codebases, since the number of XML files increases.

5.3. Gradle Trends

1. Establishment of incremental annotation processing to increase the effectiveness of incremental annotation.
2. Collaborative debugging in Gradle hugely helps in scanning of multiple scripts at one glance, reducing the processing time.
3. Supports user customization at a maximum rate.

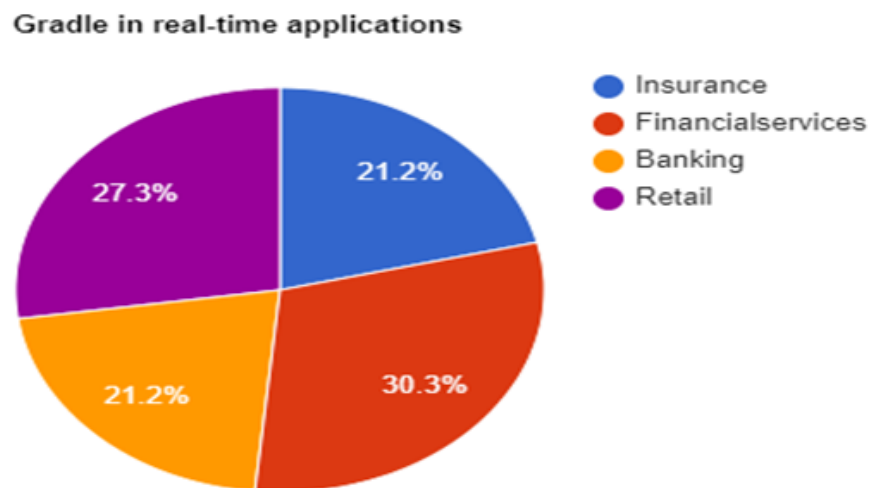


Figure 13: Gradle implementation for real-time applications [Sample size: 100] (Source Material: From References)

5.4. Gradle Challenges

1. Changes in environment variable pose a challenge for tracking it.
2. Lack of adequate documentation requirements of Groovy DSL [Dynamic structured language] to write the build configuration.

5.5. Bazel Trends

1. Supports multiple project integration with toolchains and platforms.
2. Aims to excel at supporting multiple platforms and mixed language basis of project development.
3. In architectural implications, Bazel avoids cyclic dependencies.

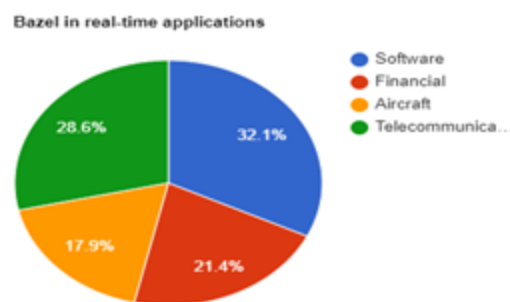


Figure 14: Bazel implementation for real-time applications [Sample size: 100] (Source Material: From References*)

5.6. Bazel Challenges

1. Explicit listing of dependencies makes the build process a tedious and a repetitive task.
2. Bazel uses sandboxing to ensure the build correctness, but sandboxing creates over head in performance.

5.7. User Analysis

It is therefore evident that Maven leads the field in terms of adoption by offering more user-friendly features for developers and programmers.

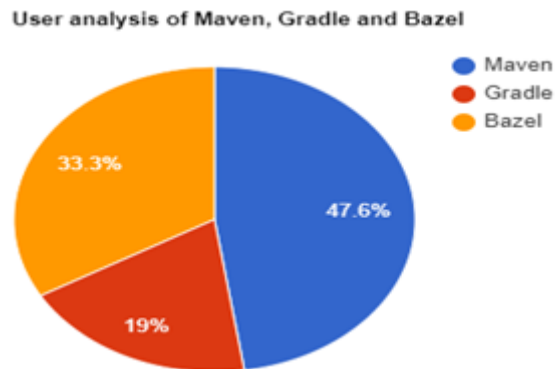


Figure 15: Usage Analysis of Maven, Gradle and Bazel Tools across industries [Sample size: 100](Source Material: From References*)

6. SUMMARY AND COMPARISON OF MAVEN, GRADLE AND BAZEL

Build Tool Critical Features	Maven	Gradle	Bazel
Open-Source	1	1	1
Java-Projects Enabled	1	1	1
Multiple Platform Support	1	1	1
Code Management Efficiency	1	1	1
Multiple Language Support	1	1	1
Build Execution	-1	1	0
Use of Xml File	1	-1	-1
Customization	-1	1	0
Ease of Understanding	0	1	1
Dependency Management	0	1	0
Total	4	8	5

Weights (-1 being weakest feature, 0 being average and 1 being strongest feature)

6.1. Build Compilation Time Analysis of Maven, Gradle, and Bazel

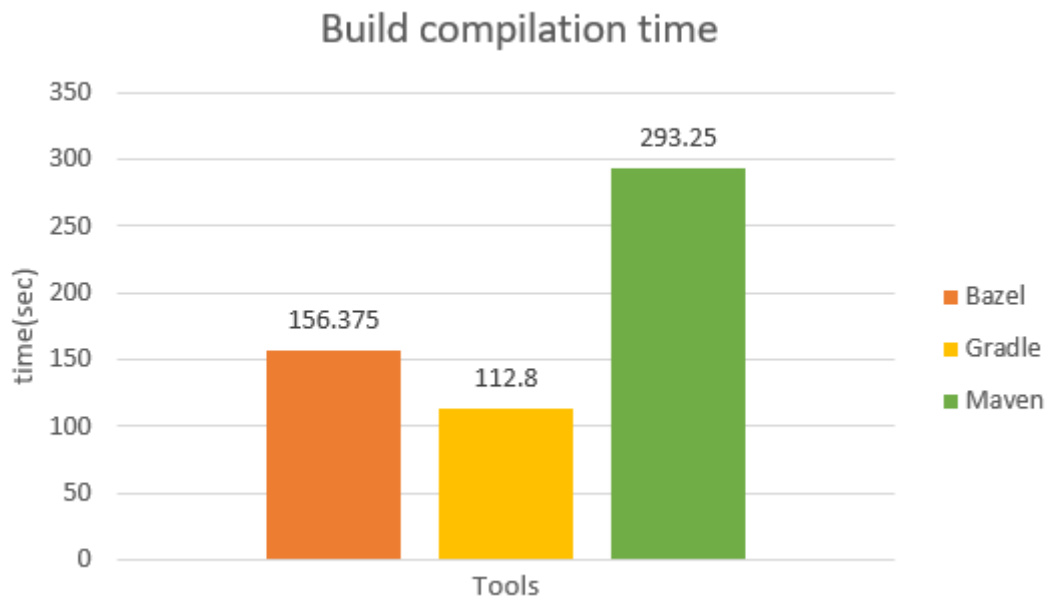


Figure 16: Bar-Graph representation of execution time analysis of Maven, Gradle and Bazel

The above Bar-Graph is drawn using the average time compilation of small, medium and large projects using Maven, Gradle and Bazel. The factors we use in a project decide the efficiency of a build system and its performance. The approach for making this comparison is based on so many features like performance, user experience, dependency management and flexibility. This graph concludes that Maven takes more time to perform the build compilation followed by Bazel and Gradle. This shows us that on an average Gradle is more effective for all types of project. Among these three build automation tools it is found that Gradle will complete the process within a short period of time, 112.8 sec (average of small, medium and large projects together). Therefore, Gradle is more effective and helpful for the developers.

7. CONCLUSION

The main purpose of this paper is to analyse and provide an overview of automation tools. Gradle and Maven are almost similar in building projects along with their dependencies, but we have seen Bazel follows different approaches.

Gradle gives best results in terms of performance and Maven is remarkable in providing an overall easy build experience for users.

Maven focus on developing a project within the deadline whereas Gradle develops the application by adding functionalities. Gradle has many advanced CLI features for effective build automation. Bazel supports IntelliJ IDEA(Java IDE) that works properly in MAC and is not supported in windows.

With reference to API's Maven and Gradle are good at providing flexible plugin provisions with the other IDEs and have created a benchmark for plugin user-facilities to a large extent. Maven provides overriding dependencies based on the version that too only on a single set of

dependencies. Whereas Gradle and Bazel both supports similar dependency management flow but Gradle has more advanced features and capabilities in dependency management.

Based on this comparison we can conclude that both Bazel and Gradle are faster than Maven. The studies suggest that Gradle is better than Bazel for most of the projects. Even if we take a million lines of code Gradle performance is fast and reliable than Bazel and Maven.

However, it is difficult to choose a single tool for project development, when all the tools have their own advantages and disadvantages. We can consider Maven while working on a small project where modularization, consistency and lots of plugins are required. Gradle is best suited for large projects, where the focus is on flexibility, speed, ease of use and integration builds. Bazel is designed purely to handle very large projects. Projects started in Maven, at any phase can be easily ported to Gradle and vice versa. Gradle can handle all the projects with utmost flexibility and speed. It also supports porting between the tools in minimum steps. Hence, Gradle can be considered as the most preferred automation tool in recent times. Future exposure should critically look at the break down of existing plugins in Gradle because this is leading to unexpected changes in build operation causing a lot of restrictions in usage of functionalities and can work on the complex dependency trees in Maven to minimize the disputes within dependencies.

ACKNOWLEDGEMENTS

The author would like to thank the organization L&T Technology Services(LTTS) for giving the opportunity to work on Build Tools. Thanks to the immense resources online on Maven, Gradle and Bazel that helped to get deeper understanding.

REFERENCES

- [1] Kaiyun Wang, Greg Tener, Vijay Gullapalli, Xin Huang, Ahmed Gad, Daniel Rall, “Scalable Build Service System with Smart Scheduling Device”, International Symposium on Software Testing and Analysis (ISSTA), July 18-20,2020.
- [2] Mubarak Albarka Umar, Zhanfang Chen, “A Study of Automated Software Testing: Automation Tools and Framework”, International Journal of Computer Science Engineering, December 2019, Volume 8.
- [3] Adrian Paschke, “OntoMaven API4KB-A Maven-based API for Knowledge Bases”, International Workshop on Semantic Web Applications and Tools for Life Sciences, December 2013.
- [4] Adrian Paschke, Ontomaven: “Maven-based ontology development and management of distributed ontology repositories.” In 9th International Workshop on Semantic Web Enabled Software Engineering (SWESE2013). CEUR workshop proceedings, 2013.
- [5] Wilfried Elmenreich, Philipp Moll, Sebastian Theuermann, Mathias Lux, “Making simulation results reproducible— Survey, guidelines, and examples based on Gradle and Docker”, Peerj Computer science,2019.

AUTHOR

Mridula Prakash has over 13 years' experience in the electronics industry, largely dedicated to embedded software. She is a senior executive with extensive experience in architecting and building embedded products in the Industrial Products domain. She is an active member in embedded system community and has been working on various technologies of microprocessors and microcontrollers, including x86, PIC, AVR, MIPS, PowerPC and ARM, developing firmware and low-level software in C/C++ on Linux, Android, FreeRTOS and many other kernel and operating systems.



In her current role as Specialist – Embedded Architect at L&T Technology Services (LTTS), she is responsible to understand major trends in the embedded sector and help in the implementation of embedded design software and modernization of legacy system.

© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

ENHANCED GREY BOX FUZZING FOR INTEL MEDIA DRIVER

Linlin Zhang and Ning Luo

Visual Computing Group,
Intel Asia-Pacific Research & Development Ltd, Shanghai, China

ABSTRACT

Grey box fuzzing is one of the most successful methods for automatic vulnerability detection. However, conventional Grey box Fuzzers like AFL can open perform fuzzing against the whole input and spend more time on smaller seeds with lower execution time, which significantly impact fuzzing efficiency for complicated input types. In this work, we introduce one intelligent grey box fuzzing for Intel Media driver, MediaFuzzer, which can perform effective fuzzing based on selective fields of complicated input. Also, with one novel calling depth-based power schedule biased toward seed corpus which can lead to deeper calling chain, it dramatically improves the vulnerability exposures (~6.6 times more issues exposed) and fuzzing efficiency (~2.7 times more efficient) against the baseline AFL for Intel media driver with almost negligible overhead.

KEYWORDS

vulnerability detection, automated testing, fuzzing, Grey box fuzzer.

1. INTRODUCTION

Grey box fuzzing is a popular and effective approach for vulnerability discovery. As opposed to black box approaches which suffer from a lack of knowledge about the application, and white box approaches which can incur high overheads due to program analysis, grey box leverage a lightweight code instrumentation approach to achieve the balance between efficiency and overheads. American Fuzzy Lop (AFL) and its variants are the most popular implementations of Grey box fuzzers.

However, several limitations may greatly impact the fuzzing efficiency for large scale software with complicated input, like Intel Media Driver. Firstly, without input structure awareness, Grey box fuzzers usually perform bit level mutation on the whole input indiscriminately which is ineffective on exploring the vast yet sparse domain of expected inputs. Secondly, they tend to spend more time on seeds with lower execution time – in real case, 99% of fuzzed inputs will be directly rejected by the input validity check and cannot enter & verify the core logic of the software.

In this work, we introduce one novel intelligent grey box fuzzing for Intel Media driver, MediaFuzzer. Which can do effective fuzzing based on selective fields of the input. As the core of MediaFuzzer, we also create one novel calling depth-based power schedule biased toward seed corpus leading to deeper calling chain and more likely to pass the parameter validity check.

Per Our evaluation, against the above two innovations, MediaFuzzer can dramatically improve the vulnerability exposures (~6.6 times more issues exposed) as well as the fuzzing efficiency (~2.7 times more efficient) comparing with its baseline AFL for Intel media driver with almost negligible extra overhead.

2. MEDIA FUZZER OVERVIEW

Media Fuzzer working flow can be shown by [Figure 1] below.

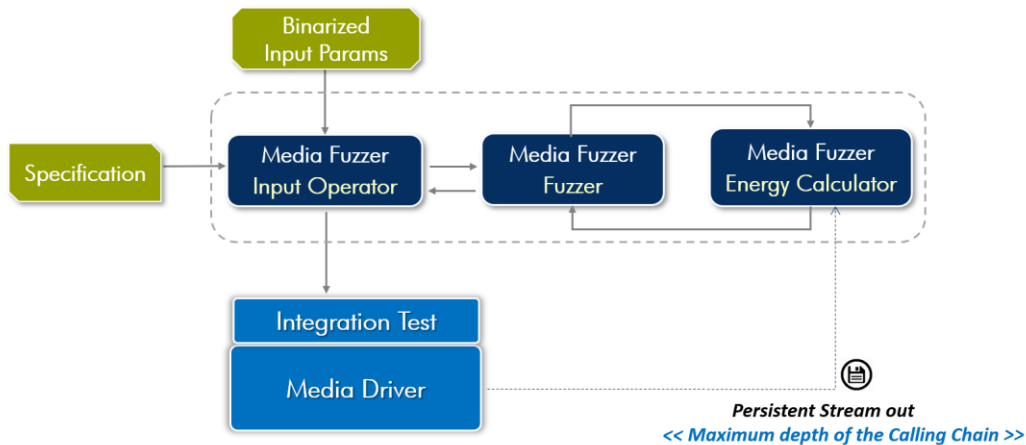


Figure 1: Media Fuzzer Working Flow

Media Fuzzer is hooked onto the input of Intel Media Integration test. The input parameters will be binarized and feed into Media Fuzzer before entering the real test case.

For each input param type, there is one accompanying specification file articulating its binary layout as well as those interesting fields to the following fuzzing operation.

Fuzzing operation will be performed against Partial of the input spliced from the aforementioned interesting bits. Input operator will parse the binarized input, extract the interesting fields and generate initial input (seed corpus), based on settings in specification file. The seed corpus generated will be used as the input to the following fuzzing pipeline.

Starting with initial input (seed corpus), Fuzzer module of MediaFuzzer, which is implemented on top of AFL core, will perform the mutation against pre-defined set of generic mutation operators (e.g., bitflips).

Energy Calculator determines how much time is spent on fuzzing operations for one seed.

In Media Fuzzer, we innovated one validity-based power schedule which tends to assign more energy to (spend more time on) inputs leading to deeper calling chain.

Based on the decision from Energy Calculator, the mutated inputs deemed sufficiently new will be mutated further to explore more inputs.

The Fuzzed input will then be pass back to Input Operator and restored to its original form and then feed into Media driver Integration test for test case execution.

Series of trace messages are added inside Intel Media Driver to record to the maximum calling chain depth (inside driver) in current execution. After each execution, one special designed postscript will help to parse the output, extract the aforementioned maximum calling chain depth data, and get it feedback to Energy Calculator for energy data update of the active seed.

Next let us put more focus on the 2 key innovations in MediaFuzzer:

- **Effective fuzzing based on Partial Input**

In contrast to conventional Grey box fuzzers, MediaFuzzer is input-structure aware which can perform bit level mutation on selective fields for effective fuzzing based on partial of the input.

As can be shown by [Figure 2] below,

Before fuzzing, all input params will be binarized and passed into Input operator module firstly. Input operator will extract those interesting fields based on the specification settings and generate the initial seed corpus for the following fuzzing operation.

After Fuzzing operation done, the fuzzed seed will be passed back to Input operator again and restored into their original form. The restored input with fuzzed bits will then be used as the input to media integration test.

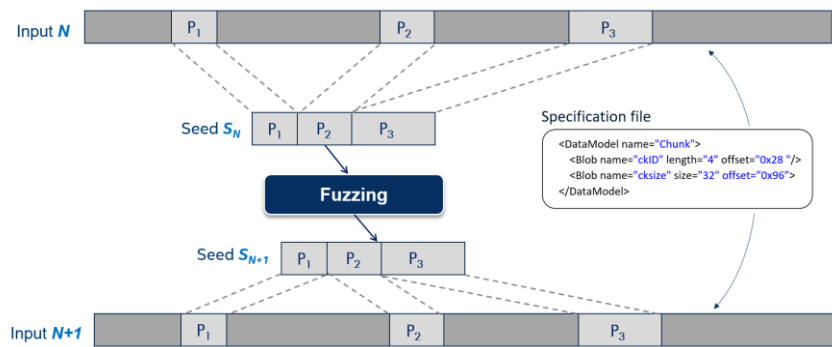


Figure 2. Enhanced fuzzing based on Partial of input

- **Power Schedule based on Calling Chain Depth**

Power schedule in fuzzer is to determine how much time should be spent on fuzzing operation for each seed. Several conventional power schedules are used before. E.g, the power schedule of AFL assigns more energy to smaller seeds with a lower execution time that have been discovered later.

For effective fuzzing on Media Driver, we introduce one novel calling chain depth-based power schedule in which each seed is assigned an energy based on the maximum depth of the calling chain in each execution (triggered by this seed). It tends to spend more time on seeds which can step deeply into the driver under test.

Some more details about the forementioned calling chain depth-based power schedule are as below:

The degree of validity $v(s)$ of a seed s is determined by the maximum calling depth of the last execution. If last time the seed lead to the historical maximum calling depth, its degree of validity $v(s) = 100\%$. If the previous calling depth is half of the historical maximum, its validity $v(s) = 50\%$.

Given the seed s , the depth-based power schedule power schedule $P_v(s)$ assigns energy as below

$$p_v(s) = \begin{cases} 2p(s) & \text{if } v(s) \geq 50\% \text{ and } p(s) \leq \frac{U}{2} \\ p(s) & \text{if } v(s) < 50\% \\ U & \text{otherwise} \end{cases}$$

where $p(s)$ is the energy assigned to s by the traditional AFL's original power schedule and U is a maximum energy that can be assigned by AFL. This power schedule implements a hill climbing metaheuristic that always assigns twice the energy to a seed that is at least 50% valid and has an original energy $p(s)$ that is at most half the maximum energy U .

In that case, the aforementioned depth-based power schedule will assign more energy to seeds with a higher degree of validity. It is expected more valid inputs can be generated from already valid inputs. It implements a hill climbing meta-heuristic where the search follows a gradient descent. A seed with a higher degree of validity will always be assigned higher energy than a seed with a lower degree of validity.

Our evaluation demonstrates, against the above 2 innovations, MediaFuzzer can dramatically increase the vulnerability exposure comparing with its baseline AFL. Within the given time limit of 12 hours, against the same test coverage, MediaFuzzer discovered 33 bugs in Intel Media Driver while its baseline (AFL) only detected 5 bugs.

Also, MediaFuzzer is more efficient than AFL.

As shown by [Figure 3] below, against the same integration test, MediaFuzzer can discover the same number of code paths in 9 hours for which AFL requires 24 hours (i.e., about $2.7\times$ more efficient).

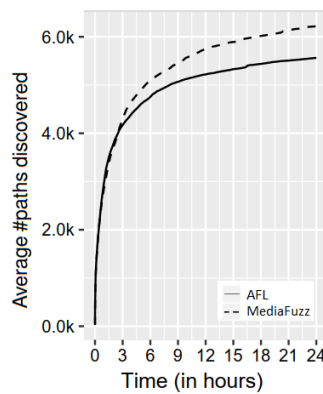


Figure 3. Average #Code Paths discovered in 24 hours for Intel Media driver

Meanwhile its extra overhead is almost negligible. With our optimization, MediaFuzzer now can achieve similar execution speeds to AFL - $\pm 2\%$ for average execution time per seeds.

3. SUMMARY

Grey box Fuzzing is a popular and effective approach for vulnerability discovery.

However, conventional Grey box fuzzers like AFL is structure unaware and tend to spend more time on smaller seeds with lower execution time, which greatly impact its fuzzing efficiency for complicated input types.

In this work, we introduce the intelligent grey box input parameter fuzzing, MediaFuzzer, for Intel Media driver. It supports effective fuzzing based on selective fields of input and creates one novel calling chain depth-based power schedule biased toward the seeds corpus leading to deeper calling chain and more likely to pass the parameter validity check.

Per Our evaluation, with the above innovations, MediaFuzzer can dramatically improve the vulnerability exposures (~6.6 times more issues exposed) and fuzzing efficiency (~2.7 times more efficient) than its baseline AFL for Intel media driver with negligible extra overhead.

We believe the similar methodology can also be applied on and benefit other Intel Software.

ACKNOWLEDGMENTS

We would like to express special thanks of gratitude to our Boss, Xiong Andy, who granted us the great opportunity for this interesting research and get MediaFuzzer applied on our media driver development.

Also, thanks to our Intern, Sun, Weiqi for your help on all kinds of data collection in our research.

REFERENCES

- [1] A. Arcuri & L. Briand, "A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering," *Softw. Test. Verif. Reliab.*, vol. 24, no. 3, pp. 219–250, May 2014.
- [2] V. J. M. Manes, H. Han, C. Han, S. K. Cha, M. Egele, E. J. Schwartz, & M. Woo, "The art, science, and engineering of fuzzing: A survey," 2018.
- [3] Y. Lian & Z. Hu, "Smarter peach: Add eyes to peach fuzzer," in *RootedCon*, 2017.
- [4] H. Peng, Y. Shositaishvili, & M. Payer, "T-Fuzz: Fuzzing by program transformation," in *IEEE Symposium on Security and Privacy (S&P)*, 2018.
- [5] T. Petsios, J. Zhao, A. D. Keromytis, & S. Jana, "SlowFuzz: Automated domain-independent detection of algorithmic complexity vulnerabilities," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.
- [6] V. Pham, M. Bohme, & A. Roychoudhury, "Model-based white-box fuzzing for program binaries," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2016.
- [7] V.-T. Pham, M. Bohme, A. E. Santosa, A. R. Caciulescu, and A. Roychoudhury, "Smart greybox fuzzing," 2018.
- [8] E. J. Schwartz, T. Avgerinos, & D. Brumley, "All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask)," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, ser. SP '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 317–331.
- [9] N. Stephens, J. Grosen, C. Salls, A. Dutcher, R. Wang, J. Corbetta, Y. Shoshitaishvili, C. Kruegel, & G. Vigna, "Driller: Augmenting fuzzing through selective symbolic execution," in *Proceedings of 23rd Annual Network and Distributed System Security Symposium (NDSS)*, 2016.

AUTHORS

Linlin Zhang is a senior software engineer at Intel. Her research interests include software architecture, quality, continuous delivery, and DevOps. Please contact her at livia.zhang@intel.com



Ning Luo is the senior software architect at Intel. Please contact him at ning.luo@intel.com.



© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

RUNWAY EXTRACTION AND IMPROVED MAPPING FROM SPACE IMAGERY

David A. Noever

PeopleTec, Inc., Huntsville, AL, USA

ABSTRACT

Change detection methods applied to monitoring key infrastructure like airport runways represent an important capability for disaster relief and urban planning. The present work identifies two generative adversarial networks (GAN) architectures that translate reversibly between plausible runway maps and satellite imagery. The training capability was illustrated using paired images (satellite-map) from the same point of view and using the Pix2Pix architecture or conditional GANs. In the absence of available pairs, the CycleGAN architecture likewise showed that its four network heads (discriminator-generator pairs) also provided effective style transfer from raw image pixels to outline or feature maps. To emphasize the runway and tarmac boundaries, the experiments show that the traditional grey-tan map palette is not a required training input but can be augmented by higher contrast mapping palettes (red-black) for sharper runway boundaries. The research highlights a potentially novel use case (called “sketch2satellite”) where a human sketches the current runway boundaries and automates the machine output of plausible satellite images. Finally, faulty runway maps were identified where the published satellite and mapped runways disagree, but an automated update renders the correct map using GANs.

KEYWORDS

Generative Adversarial Networks, Satellite-to-Map, Pix2Pix, CycleGAN Architecture.

1. INTRODUCTION

Airport runways present an attractive challenge for overhead object recognition and modern machine learning. Urban planning, disaster relief, and air safety benefit from tracking runway changes over time (e.g., modeling change detection). In human terms, more than half of all airline accidents occur near airports during take-off, approach, and landing [1]. Urban planners (such as the International Airports Council) cite airports as the central hub of decaying global infrastructure, with an estimated five-year budget requiring greater than \$128 billion investment in the US alone [2]. A better understanding of current runway condition, either from space or drone imagery, highlights the need for continuous monitoring much like as has been done for highway maps [3]. Therefore, one motivation for the current work is to automate the reversible transformation of unmarked overhead imagery with map-like outlines showing the most current conditions of runways, flight lines, tarmacs, and aprons. In other words, given any runway image, design a method to generate the corresponding map pair and vice versa, as illustrated in Figure 1.

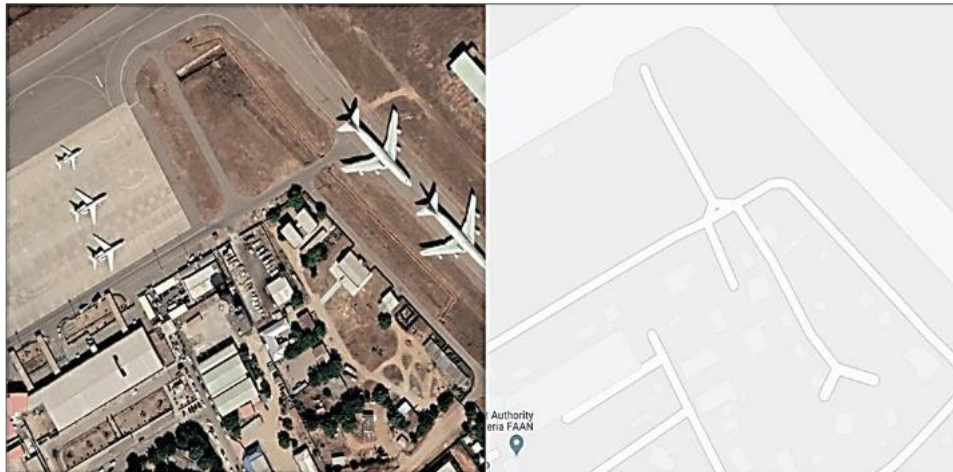


Figure 1. Example Runway Image and Map Pair. Mallam Aminu Kano International Airport, Kano, Nigeria (ICAO Code DNKN).

1.1. Motivation

A global map identifying the 45,000 small and large places to land an airplane, effectively mirrors the entire earth's landmass with recognizable terrain features (Figure 2, adapted from [4]). However, in contrast to the extensive highway and roadway analysis [5], the details of corresponding runway maps have traditionally been low, often just showing a single set of parallel lines to signify the runway. Details like runway damage that might otherwise be identified from updated satellite surveys have not offered a corresponding way to update or evaluate map changes. Sustainable airport management requires real-time evaluation methods, particularly for smaller airports or military bases [6]. Deep learning methods have shown that aerial (drone) imagery of runway cracks could automate key aspects of inspections [7]. Other cited reasons for not examining the runway integrity problem stem from obfuscation strategies (see examples, [8]). Notably, the combination of military and civilian runways in entire countries (like Greece) has prompted Google to provide few details in either imagery or maps. Many satellite and map publishers have adopted policies to obfuscate either the image or the map, depending on either the country's or owner's request. Other motivations include algorithmic removal of non-stationary objects like planes from multiple images (effectively emptying the scene of context). In many of these cases, the map is available in full resolution, but the satellite imagery is pixelated beyond recognition (e.g., see Kos International Airport, 36.801622, 27.089944)

1.2. Execution of Previous Work

For mapping airport runways algorithmically, the present work examines two popular strategies: Pix2Pix and other generative adversarial networks (GAN) models [9] like CycleGANs. Other image-to-image methods generate synthetic data using CycleGANs [10], Deep Priors [11], Pix2Pix [9], and PatchGANs [12]. The research addresses the popular image-to-image (I2I) translation as one concrete example with a practical application [13-14], namely the correction of flawed airport runway maps. The work generates and explores a novel dataset for this specialized problem, the translation of an arbitrary airport satellite image into a plausible runway map [15]. By specializing in a single feature like airport runways, one key assumption ignores the complexities of previous approaches that have combined urban landmarks; ignoring these buildings, roads, and terrain features are tested against the hypothesis that better output might emerge from the more singular focus on one class [16]. The research hypothesizes that a

dominant single object type (such as runways) may improve predictive performance. This research systematically controls the map contrast and color saturation to provide new methods for data augmentation and visualization beyond the traditional grey-tan color maps by creating higher contrast black-red maps.



Figure 2. Global airport distribution (Davenport, 2013). Each dot represents an airport and total 45,132 runways ranging from mega-hubs to single dirt roads. The image captures the land borders and global population density using just airport runways (Davenport, 2013).

Pix2Pix models have solved several interesting generator-discriminator problems for paired images, including automating the conversion of satellite photos to road maps and vice versa (e.g., sat2map and map2sat transformation, see [9]). As examples of image-to-image translation and conditional GANs, or cGANs, the output image production depends on the input or paired image. Training urban road data has previously relied on collecting and pairing satellite imagery with low-contrast Carto DB maps [17]. These pairs have included complex mixtures of objects, primarily focused on urban scenes with buildings, roads, and parks all combined in the same image. Recent work [16] suggested that research improvements should narrow the object ontology and test for better aesthetic performance.

In addition to narrowing the ontology for runways only, one secondary goal here sought to improve the map contrast examples from the diverse field of GANs include using twin neural networks to spoof satellite images for fake archives and fuse the broad spectral platforms now coming online. The present work applies CycleGANs as one alternative method to generate plausible synthetic runway maps but without relying on precisely paired imagery as conditional requirements. These methods achieve a style transfer between raw images and outline maps as two curated collections without matching location or precise times.

1.3. Original Contributions

The research effort offers a novel dataset for training and testing style transfer algorithms like Pix2Pix and CycleGAN. Unlike previous urban settings with mixed object classes, the runway example presents a more uniform case to compare and contrast the algorithmic output. Runways also offer a high-value infrastructure monitoring example where out-of-date or incomplete imagery and maps are not uncommon. The approach modifies the color palettes of both input and output maps (low-to-high contrast) to examine changes in conditional outputs and enhance visual quality. The research identifies examples where either the image or map do not match in existing ground truth by human analysis, then compares the generative models for improving on the status quo for incomplete runway mapping. The paper finally treats the novel use case of converting sketches and rough maps to generate plausible satellite maps. beyond the light grey Carto DB palette and see if a wider color range can improve the Pix2Pix outcome. To explore these issues,

the work created a novel dataset of 2400 airport locations, mapped in tandem with both Google satellite and map images combined. The research systematically enhanced the map color contrast to reduce artifacts and improve appearance for both the “sat2map” and “map2sat” cases.

2. METHODS

For understanding automated runway maps from images and vice versa, the work assembled two datasets. The first one, used for Pix2Pix, requires the map rendering to pair with the exact satellite image in the same location and time. The conditional GAN learns the style transfer features from one to the other, similar to what a colorizing or skeletonizing function might attempt. A Pix2Pix example transforms night and day imagery in pairs. The second dataset, used for CycleGAN, requires no exact pairing of maps and images but instead needs a collection of representative examples for both. The popularity of CycleGAN for doing this kind of unpaired training motivates some classic examples where natural pairs would not normally exist (e.g., horse-to-zebra transformation is not naturally available in the same poses). The primary architectural differences stem from the 2 networks (discriminator-generator) for Pix2Pix but 4 for CycleGANs.

2.1. Dataset Construction and Modification

The dataset was created from image pairs of the same location in both map and satellite imagery using the global airport database (9300 latitude-longitude locations, [15]). The airports ranged across all continents with the highest concentration in the US (552) and Germany (529). The assembled 2500 images (1200x1200 pixels) were labeled using Google Map API both in satellite and map modes of each location centered by latitude and longitude with zoom 18 (approximately 800-1000 feet in altitude). Each square image covers approximately 0.3 square miles. Each image was labeled with the four-letter airport code as location indicators (ICAO, International Civil Aviation Organization).

2.2. Dataset Construction and Modification

The runway imagery and maps were scaled to 600x600 pixels, then joined as pairs (1200x600) for Pix2Pix training and validation (Figure 1). Batch image transformations were done using ImageMagick [18]. For CycleGAN training, the images were reduced further (256x256 pixel) and not joined as pairs but divided into separately labeled folders for A-B and B-A transformations during training (e.g., map-satellite, satellite-map). Infrastructure maps have used drone aerial imagery and noted the technical challenges of inaccurate or incomplete training data [19]. This method avoids some of these by relying heavily on the synchrony between satellite and maps already within Google Map API. Where obvious differences were noted between the map and image (either because of ground truth changes or poor inputs), the research collected those mismatches into inference cases to consider post-training as example use cases. The overlay of place names on maps proved unavoidable based on the API collection method and no attempt was made to remove or obscure them in training or inference datasets.

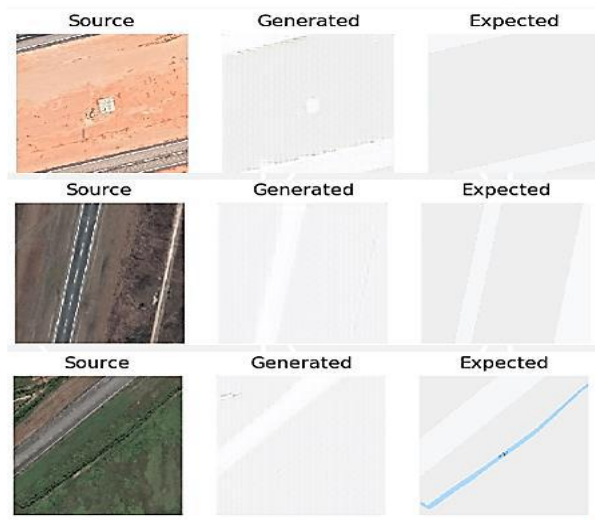


Figure 3. Pix2Pix model output of plausible maps from satellite images. Generated maps are shown compared to ground truth or expected output from Google Maps API.

2.3. Training Approach

For both Pix2Pix and CycleGAN methods, the model building benefited from the Keras library for deep learning [20-21]. The Pix2Pix training for discriminator (D) and generator (G) combinations used the following hyper-parameters: batch size = 1, epochs = 10, base image size = 600x600 pixels in paired A-B sets (600x1200). The CycleGAN training for two sets of D-G combinations used the following hyper-parameters: batch size = 1, epochs = 100, base image size = 256x256 pixels in unpaired A-B sets. The compiled CycleGAN model used the stochastic gradient descent or Adam optimizer (learning rate=0.0002, beta1=0.5, loss_weights=0.5) with mean square error (MSE) loss function. All training runs were conducted using a graphical processing unit (GPU model RTX 5000) with a comparative compute capability=7.0 [22]. Even when trained on relatively powerful GPU capabilities, the models require several days of training each on dedicated hardware.



Figure 4. Conditional GAN (Pix2Pix) model for translating maps into plausible satellite imagery. The generated example show learning progress over a number of iterations, with later examples showing more texture and color details.

3. RESULTS

Using the trained GAN models, inference steps ideally generate plausible runway maps from unlabelled satellite images and vice versa. Figure 3 summarizes an example Pix2Pix output of maps generated from satellite imagery.

3.1. Map-to-Satellite (Pix2Pix Learning)

For conditional GANs with a required input and an expected outcome (such as Pix2Pix), Figure 4 shows an example output of plausible satellite imagery generated from a simple input runway map. Over many learning steps, the generated satellite imagery includes more texture and color details but does not increase overall realism.

3.2. Single Class Comparison

For comparison to traditional multi-class Pix2Pix, Figure 5 shows both a complex urban scene trained to highlight mapped roads and a single-class airport scene trained to show runways. This result supports the hypothesis [16] that reducing the complexity or number of classes in the satellite image might improve map generation. The finer details of roads in Figure 5 appear lost compared to the original satellite image with background buildings and foliage. The simpler runway image generates a reasonable map of coarse but accurate features as needed for navigation or status assessments.



Figure 5. Comparison of multi- and single-class Pix2Pix generated maps. The urban satellite image generates a lower resolution map lacking the features expected and the single-class map shows the simple runway outline as expected.

3.3. CycleGAN Results

Figure 6 shows an example output from both the map and runway inputs. The translated images and map capture a reasonable case, but the maps from CycleGAN generally show less detail and artifacts compared to the maps generated by Pix2Pix paired images. The map-to-satellite transform however performs comparably to Figure 4, Pix2Pix generation.

3.4. CycleGAN Results

For qualitative comparison, the CycleGAN maps appear less convincing in Figure 7 compared to Pix2Pix output. The satellite images for CycleGAN (from map inputs) however show fine details including expected runway color and texture for foliage and open spaces. Although both methods offer reversible translations (map-satellite and satellite-map) the desired runway map features appear superior in Pix2Pix outputs. Depending on the use case, the CycleGAN's ability to generate meaningful color palettes (green) and detailed foliage in "fake" satellite imagery may serve for more lucid descriptions of runway changes.

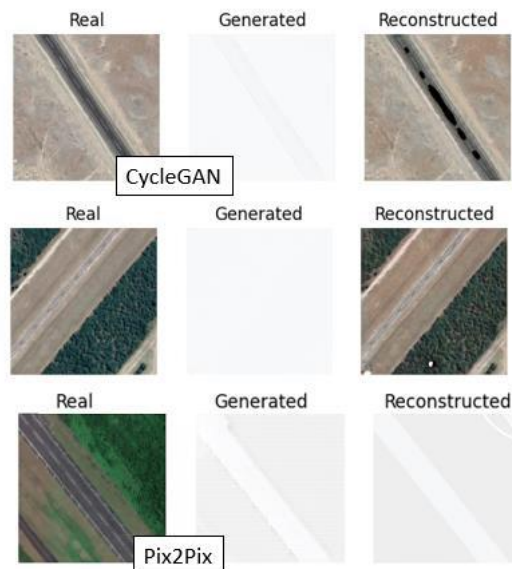


Figure 6. Comparative output for CycleGAN's four networks vs. Pix2Pix conditional GAN use of 2 networks. The satellite-to-map translation for CycleGAN (rows 1-2) loses observable details but compares favorably during image reconstruction. The Pix2Pix model (paired images) provides superior maps.

3.5. High-contrast Map Results

To test whether higher contrast maps can generate better satellite imagery (or vice versa), the input maps were colorized from the dominant tan-grey palette used by Google Maps API to a red-black palette. The CycleGAN model results as trained on high contrast maps are illustrated in Figure 7. While a human geographer might find the high-contrast maps less satisfying, the features of particular interest like runway boundaries, tarmacs, and open space are delineated by high-contrast. The algorithms, both Pix2Pix and CycleGAN, are trainable on any map color palette desired for best results.

3.6. Sketch Map Runways

For the interesting use case of a human observer who inspects runway status, then sketches the infrastructure's overall condition, Figure 8 illustrates the image-to-image translation capabilities. The simple maps provide a corresponding plausible satellite image pair. The geometric details of various cross and loop patterns get textured by the Pix2Pix translation. This application represents the first instance of “sketch2satellite” which can be compared to older ground-truth satellite imagery and spawn the corresponding change detection imagery based on human observers, social media posters, and first responders.

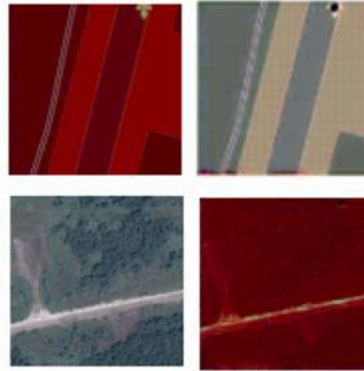


Figure 7. High-contrast maps (red-black) examples for both A-B and B-A transformation with map2sat (top) and sat2map (bottom). Finer foliage details are seen in the high contrast maps for the bottom row compared to the standard tan-gray palette.

3.7. Faulty Runway Corrections

A host of faulty runway maps were identified in the development of the airport dataset. For the sake of clarity, the satellite map generally was assumed to be ground truth, such that running inference models plausibly corrects extra runways or tarmacs misidentified in other public datasets such as Google Maps API. The reverse problem would assume the map is correct and plausibly generate a corresponding satellite image. One implication of this approach is to automate the change detection of outdated maps using recently downloaded orbital images. This change detection scenario effectively automates the most current map always being available for infrastructure monitoring and planning. This similarly has attributes that could be exploited for evaluating runway damage or expected repairs using remote sensing alone.

3.8. GAN Artifacts

A well-known set of artifacts have been noted from previous GAN literature, including smudges, unnatural scenery, and phantom landmarks. Figures 3-11 show examples of artifacts that would alert a human expert that the generated image or map is a “fake” or machine generated version compared to the expected ground truth. However, unlike GAN examples with fake faces or counterfeit objects, the generation of runway descriptions (either satellite or maps) adds to the available information and the detectable authenticity proves subordinate to the original intentions. A fake map is assumed to be a simplified overhead image, much like a fake cartoon or painted portrait would be understood as not representing an authentic face.



Figure 8. CycleGAN runway2map and map2runway examples. The bottom row is generated from input examples in the top row.

4. DISCUSSION

In addition to curating a novel runway dataset, the present research has 1) demonstrated that single-class image translations can yield lucid maps that highlight the key features like runway boundaries compared to more complex urban environments; 2) identified previously published but incorrect maps using up-to-date satellite imagery to spawn corrected runway boundaries; 3) proposed novel use cases where high-contrast maps or hand-sketches can generate plausible satellite imagery *ab initio* without requiring paired ground truth for predictions.

5. FUTURE WORK

An important trend in machine learning has enlarged neural networks, first in-depth with single to multiple layers, then in number or “heads” with single networks to competing ones, and finally in domain expertise with attention and specialized training fields. A key development of multiple networks was the generator-discriminator paradigm that led to generative adversarial networks (GANs). Their remarkable abilities to learn and extend the traditional function approximation or pattern recognition present more creative tasks. Future work builds on and draws inspiration from the rapid growth of small satellites and their concomitant low cost, rapid revisit rates, and public data subscriptions.

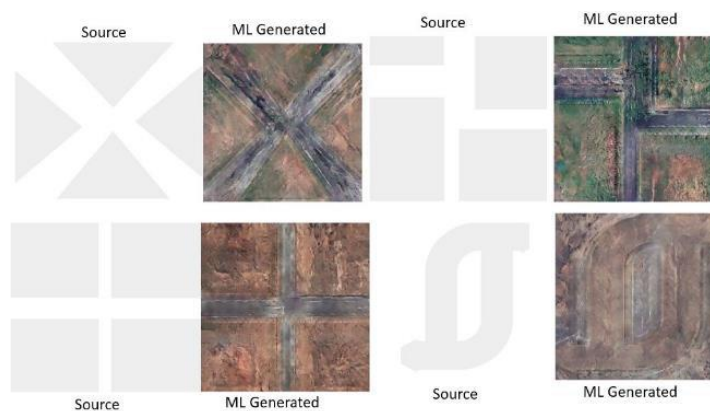


Figure 9. Example sketched maps and the corresponding generated satellite images. The Pix2Pix model renders plausible synthetic satellite data from hand sketches for runways.



Figure 10. Faulty Google Map API runway map for Lae Nadzab Airport, Papua New Guinea (AYNZ). The satellite image shows a three-way intersection, which is correctly captured by the Pix2Pix model but incorrectly shown in the published map.

The research specialization by the spectral quality has broadened considerably with new data fusion opportunities [23]. The recent XBD data competition has examined the before-and-after pairings for estimating disaster damage for different grades of building loss [24]. Compared to building assessments, the status of re-routing traffic from airports has taken on significance for example in recent relief efforts in Haiti. Further inspiration has followed for the energy industry (pairing EO/IR for estimating thermal efficiency), insurance (damage estimation), and urban planning (satellite to map). Another key development that is just emerging is the multi-domain or multi-modal generalization, where different sensory inputs or perspectives on the same event or object merge into a more interesting and complete picture. There are strong biological analogies for this sensor fusion initiative but the ability to guess a face from a voice or estimate an author from a text snippet represents innovative opportunities for blending multiple domains. Most broadly, if computers achieve expert-level human capabilities for sight, sound and reading, what are the opportunities with creative combinations? A practical implementation step might begin to use the multi-layer, multi-headed networks described here to bridge ever more elaborate creative tasks, sometimes referred to as hallucinating probabilities for sensory translation problems. For Pix2Pix models specifically, the alignment of optical objects like a building with its radar signature (synthetic aperture radar, SAR) offers one intriguing model.

ACKNOWLEDGMENTS

The author would like to thank the PeopleTec Technical Fellows program for its encouragement and project assistance.

REFERENCES

- [1] Jackman, F. (2014) “Nearly Half of Commercial Jet Accidents Occur During Final Approach, Landing”, FlightSafety Foundation, <https://flightsafety.org/asw-article/nearly-half-of-commercial-jet-accidents-occur-during-final-approach-landing/>
- [2] Baldwin, S. (2019) “Here’s why decaying US airports are turning to private money”, CNBC, Oct 2, 2019. <https://www.cnbc.com/2019/10/02/why-us-airports-are-so-bad.html>
- [3] Bello-Salau, H., Aibinu, A. M., Onwuka, E. N., Dukiya, J. J., & Onumanyi, A. J. (2014, September). Image processing techniques for automated road defect detection: A survey. In 2014 11th International Conference on Electronics, Computer and Computation (ICECCO) (pp. 1-4). IEEE.
- [4] Davenport, J. (2013), “The World, Traced by Airport Runways”, <https://ifweassume.blogspot.com/2013/06/airports-of-world.html> including dataset from <https://ourairports.com/data/>
- [5] Chen, Z., Zhang, Y., Luo, Y., Wang, Z., Zhong, J., & Southon, A. (2021). RoadAtlas: Intelligent Platform for Automated Road Defect Detection and Asset Management. arXiv preprint arXiv:2109.03385.

- [6] Kovačič, B., Doler, D., & Sever, D. (2021). Innovative Business Model for the Management of Airports in Purpose to Identify Runway Damage in Time. *Sustainability*, 13(2), 613.
- [7] Jiang, L., Xie, Y., & Ren, T. (2020). A deep neural networks approach for pixel-level runway pavement crack segmentation using drone-captured images. *arXiv preprint arXiv:2001.03257*.
- [8] Wikipedia (accessed 2021), List of satellite map images with missing or unclear data https://en.wikipedia.org/wiki/List_of_satellite_map_images_with_missing_or_unclear_data
- [9] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).
- [10] Dou, H., Chen, C., Hu, X., Jia, L., & Peng, S. (2020). Asymmetric CycleGAN for image-to-image translations with uneven complexities. *Neurocomputing*, 415, 114-122.
- [11] Wang, X., Yan, H., Huo, C., Yu, J., & Pant, C. (2018, August). Enhancing Pix2Pix for Remote Sensing Image Classification. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 2332-2336). IEEE.
- [12] Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*.
- [13] Wang, X., Yan, H., Huo, C., Yu, J., & Pant, C. (2018, August). Enhancing Pix2Pix for Remote Sensing Image Classification. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 2332-2336). IEEE.
- [14] Varia, N., Dokania, A., & Senthilnath, J. (2018, November). DeepExt: A Convolution Neural Network for Road Extraction using RGB images captured by UAV. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1890-1895). IEEE.
- [15] Partow, A. (2017) "The Global Airport Database", <https://www.partow.net/miscellaneous/airportdatabase/index.html>
- [16] Xu, C., & Zhao, B. (2018). Satellite Image Spoofing: Creating Remote Sensing Dataset with Generative Adversarial Networks (Short Paper). In *10th International conference on geographic information science (GIScience 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [17] Kang, Y., Gao, S., & Roth, R. E. (2019). Transferring multiscale map styles using generative adversarial networks. *International Journal of Cartography*, 5(2-3), 115-141.
- [18] Still, M. (2006). *The definitive guide to ImageMagick*. Apress.
- [19] Zhang, R., Albrecht, C., Zhang, W., Cui, X., Finkler, U., Kung, D., & Lu, S. (2020, August). Map generation from large scale incomplete and inaccurate data labels. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2514-2522).
- [20] Chollet, F. (2018). Keras: The python deep learning library. *Astrophysics Source Code Library*, ascl-1806.
- [21] Brownlee, J. (2019). *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery.
- [22] NVIDIA Specification GPU, Compute Capability, 2021,
- [23] Lindén, J., Forsberg, H., Haddad, J., Tagebrand, E., Cedernaes, E., Ek, E. G., & Daneshtalab, M. (2021, October). Curating Datasets for Visual Runway Detection. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)* (pp. 1-9). IEEE.
- [24] Weber, E., & Kané, H. (2020). Building disaster damage assessment in satellite imagery with multi-temporal fusion. *arXiv preprint arXiv:2004.05525*.

AUTHORS

Short Biography

David Noever has 31 years of research experience with NASA and the Department of Defense in machine learning and data mining. He received his Ph.D. from Oxford University, as a Rhodes Scholar, in theoretical physics and B.Sc. from Princeton University, summa cum laude, and Phi Beta Kappa. His primary research interests center on machine learning, algorithms, data analytics, artificial intelligence, and novel metric generation.



© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

DAG-WGAN: CAUSAL STRUCTURE LEARNING WITH WASSERSTEIN GENERATIVE ADVERSARIAL NETWORKS

Hristo Petkov¹, Colin Hanley² and Feng Dong¹

¹Department of Computer and Information Sciences,
University of Strathclyde, Glasgow, United Kingdom

²Department of Management Science,
University of Strathclyde, Glasgow, United Kingdom

ABSTRACT

The combinatorial search space presents a significant challenge to learning causality from data. Recently, the problem has been formulated into a continuous optimization framework with an acyclicity constraint, allowing for the exploration of deep generative models to better capture data sample distributions and support the discovery of Directed Acyclic Graphs (DAGs) that faithfully represent the underlying data distribution. However, so far no study has investigated the use of Wasserstein distance for causal structure learning via generative models. This paper proposes a new model named DAG-WGAN, which combines the Wasserstein-based adversarial loss, an auto-encoder architecture together with an acyclicity constraint. DAG-WGAN simultaneously learns causal structures and improves its data generation capability by leveraging the strength from the Wasserstein distance metric. Compared with other models, it scales well and handles both continuous and discrete data. Our experiments have evaluated DAG-WGAN against the state-of-the-art and demonstrated its good performance.

KEYWORDS

Generative Adversarial Networks, Wasserstein Distance, Bayesian Networks, Causal Structure Learning, Directed Acyclic Graphs.

1. INTRODUCTION

Discovering causal relationships yields new scientific knowledge. Causal discovery involves the process of learning structures of Bayesian Networks (BN) from data. BNs are considered to be one of the most powerful models for causal inference [1]. They are graphical models representing variables and their conditional dependencies in the form of directed acyclic graphs (DAG).

However, one of the major challenges associated with causal structure learning arises from its combinatorial nature. Since increasing the number of variables, resulting in a super-exponential increase of possible DAGs, makes the problem computationally intractable, it is often not possible to perform a complete combinatorial search. Nevertheless, over the last few years, several approaches have been proposed to overcome the NP-hardness of finding DAGs [2], including the score-based methods, constraint-based methods and continuous optimization. These have achieved a varying degree of success.

Score-based methods (SBM) reformulate the combinatorial structure learning approach into the optimization of a score function accompanied by a combinatorial constraint for acyclicity. They

rely on the utilization of general optimization techniques across the combinatorial search space to discover the DAG structure by optimizing the score function. However, as the complexity of the search space remains super-exponential, additional structure assumptions and approximate searches are often needed. Constraint-based methods (CBM) utilize conditional independence tests to find the DAG structure by identifying the connections between dependent variables. Also, there have been attempts to combine the score-based and constraint-based approaches. One notable product of this idea is the MMHC [3] algorithm which uses Hill Climbing as the score function and an algorithm called Min-Max Parents and Children (MMPC) to check for relationships between the variables.

A recent breakthrough by Zheng et al. [4] utilizes a new acyclicity constraint to transform the problem from combinatorial to continuous optimization, which can be efficiently solved by conventional optimization methods. However, their original algorithm works only with linear data and does not support discrete data. Yu et al. [5] further expand on Zheng's work by developing a model named DAG-GNN based on the variational auto-encoder architecture. It also reformulates the acyclicity constraint from Zheng et al. [4] to allow for more efficient computation. In addition, their method handles both linear, non-linear continuous and categorical data. Similarly, another model named GraN-DAG [6] has also been proposed to use neural networks together with the acyclicity constraint to handle both linear and non-linear continuous and discrete/categorical data for causality learning.

Our work is focused on the use of generative adversarial networks (GANs) for causal structure learning. GANs have been successfully applied to generating synthetic images by minimizing the difference between synthetic and real data with distance metrics. They have also been experimented with to synthesize tabular data [7]. Recently, Gao et al. [8] developed a GAN-based model (DAG-GAN) that learns causal structures from data by using a Maximum Mean Discrepancy (MMD) based score function.

To leverage GANs for causal structure learning, a fundamental question is whether the data distribution metrics involved in GANs can facilitate causal structure learning. Correspondingly, this work has investigated Wasserstein GAN (WGAN) in the context of learning causal structures from tabular data. The Wasserstein distance metric from optimal transport distance [9] is an established metric that preserves basic metric properties [10-13], which has led to the Wasserstein GAN (WGAN) to achieve significant improvement in training stability and convergence by addressing the vanishing gradient problem and partially removing mode collapse [14]. However, to the best of our knowledge, so far no study has been conducted to experiment with WGAN for causality learning.

The proposed DAG-WGAN is based upon the combination of an auto-encoder and WGAN-GP by incorporating a critic (discriminator) that is designed to measure the Wasserstein distance between the real and synthetic data, together with the acyclicity constraint from Yu et al. [5]. This combination allows us to compare the performance of DAG-WGAN with other relevant models that do not involve WGAN in order to test the hypothesis about the Wasserstein metric, namely whether the involvement of the Wasserstein metric can help causal structure learning in a generative process that learns how to realistically generate synthetic data. With the explicit modelling of learnable causal relations of DAGs in the model architecture, the model learns how to generate synthetic data by simultaneously optimizing the causal structure and the model parameters via end-to-end training.

Our experiments show that the new model performs better than other models when presented with a large data variable size. In particular, the causal graphs learned by using DAG-WGAN are more accurate in higher dimensions compared to those produced by other models and the quality

of the generated data is higher than that produced from other data generating models. We demonstrate the capabilities of our model on multiple data types (linear, non-linear, continuous and discrete). The model works well with both continuous and discrete data while being capable of producing less noisy and more realistic data samples.

2. RELATED WORK

DAGs lie at the centre of causal structure learning. They consist of nodes (variables) and directed edges (connections) between the nodes, which are interpreted as direct causal relationships between the variables. If a DAG entails conditional independencies of the variables in a joint distribution, the faithfulness condition allows us to recover the DAG from the joint distribution [1]. In causal structure learning, we learn DAGs from data distributions that are exhibited with data samples.

2.1. Constraint and Score based DAG Learning Approaches

There are three main approaches for learning DAGs from data, including the constraint-based, score-based and hybrid approaches.

Constraint-based search methods create graph structures by running local independence tests to manually constrain the search space [15]. Examples of constraint-based algorithms include Causal Inference (CI) [16] and Fast Causal Inference (FCI) [17-18]. However, typically these methods only lead to equivalence classes, namely, a set of candidate causal structures that satisfy the same conditional independencies. Hence, the causal information in the output is not complete. Score-based methods use a score function to measure how well different graphs fit the data in order to identify the right causal structure based on the scores. Typical score functions include Bayesian Gaussian equivalent (BGe) [19], Bayesian Discrete equivalent (BDe) [20], Bayesian Information Criterion (BIC) [21], Minimum Description Length (MDL) [22]. As the search space is often intractable, additional assumptions about the DAGs must be made - the most commonly used ones are bounded tree-width [23], tree-structure [24] and sampling-based structure learning [25-27].

Hybrid methods use a mix of score-based and constraint-based methods to learn DAGs. One such model named Max-Min-Hill-Climbing combines constraint-based modelling and search-based learning for more accurate DAG results [3]. Another example is RELAX [28], which introduces “constraint relaxation” of possibly inaccurate independence constraints of the search space.

2.2. DAG Learning with Continuous Optimization

Recently, a new approach named DAG-NOTEARS was formulated by Zheng et al. [4], which transforms the causal graph structure learning problem from its combinatorial nature into a continuous optimization framework. The success of this method facilitates the usage of conventional optimization solvers for causal structure learning. However, the DAG-NOTEARS model has limitations in handling non-linear data. Also, it only supports continuous data.

New solutions for causal structure learning have been developed recently based on DAG-NOTEARS. Yu et al. [5] developed DAG-GNN, which performs causal structure learning by using a Variational Auto-Encoder architecture. Their model extends the capabilities of DAG-NOTEARS as it works with linear and non-linear continuous and discrete data. GraN-DAG proposed by [6] is another extension from DAG-NOTEARS [4] to handle non-linear data by learning causal relations between the variables using neural networks. The calculation of the

neural network weights is constrained by the acyclicity constraint between the variables. The model makes very few assumptions about the data and variables and can generalize well. In addition, the model can work with both continuous and discrete data. Meanwhile, according to the latest work DAG-NoCurl from [29], it is also possible to learn DAGs without explicit DAG constraints.

Notably, DAG-GAN proposed by [8] is one of the latest works that uses GAN for causal structure learning. The work involves using Maximum Mean Discrepancy (MMD) in its loss function. The resulting model handles multiple data types (continuous and discrete) However, their experiments have only covered up to 40 nodes in the graphs.

3. CAUSAL STRUCTURE LEARNING WITH DAG-WGAN

This section provides an overview of the DAG-WGAN model architecture, together with the details of its loss functions and model training. We cover both continuous and discrete data types. The DAG-WGAN model involves causal structure in the model architecture by incorporating an adjacency matrix under an acyclicity constraint - see Figure 1. The model has two main components: (1) an auto-encoder which computes the latent representations of the input data; and (2) a WGAN which consists of a critic to synthesize the data with adversarial loss. The decoder of the auto-encoder is also used as the generator in the WGAN to generate synthetic data. The encoder is trained with the reconstruction loss while the decoder is trained according to both the reconstruction and adversarial loss. The joint WGANs and auto-encoders are motivated by the success of the combination of a variational auto-encoder (VAE) with GAN to better capture data and feature representation [30].

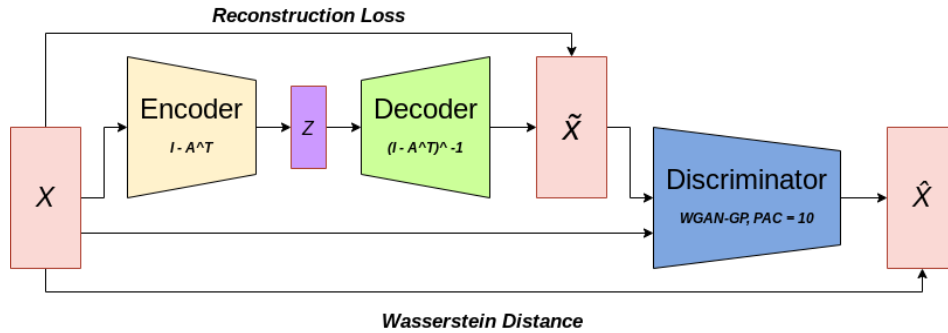


Figure 1. DAG-WGAN Model Architecture

3.1. Auto-encoder (AE) and Reconstruction

The encoder produces latent representations of the data and the decoder reconstructs the data. The representations are regularized to prevent over-fitting. The use of the auto-encoder makes sure that the latent space contains meaningful representations as the noise input to the generator in the adversarial loss training.

Similar to Yu et al. [5], we explicitly model the causal structure in both the encoder and decoder by using the structural causal model (SCM). The encoder *Enc* is as follows:

$$Enc \equiv Z = (I - A^T)f_1(X) \quad (1)$$

where f_1 is a parameterized function to transform X , $X \in \mathbb{R}^{m \times d}$ is a data sample from a joint distribution of m variables in d dimensions. $Z \in \mathbb{R}^{m \times d}$ is the latent representation. $A \in \mathbb{R}^{m \times m}$ is the weighted adjacency matrix. The corresponding decoder Dec is as follows:

$$Dec \equiv X = f_2((I - A^T)^{-1}Z) \quad (2)$$

where f_2 is also a parameterized function that conceptually inverses f_1 . The functions f_1 and f_2 can perform both linear and non-linear transformations on Z and X . Each variable corresponds to a node in the weighted adjacency matrix A .

The AE computes the latent representations through the reconstruction loss. The reconstruction loss term is defined as:

$$L(x, x') = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^d (X_{ij} - (M_X)_{ij})^2, \quad (3)$$

where M_X is a product of the decoder.

To avoid over-fitting, a regularizer is added to the reconstruction loss. The regularizer loss term takes the following form:

$$regularizer = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^d (M_Z)_{ij}^2 \quad (4)$$

where M_Z is the output of the encoder.

3.2. Wasserstein Generative Adversarial Network

The decoder of the auto-encoder is also used as the generator of the WGAN model. Alongside the auto-encoder, we use a critic to provide the adversarial loss with gradient penalty. It is based upon the popular PacGAN [31] framework with the aim of successfully handling mode collapse and is implemented as follows:

$$\hat{X} = MLP(\tilde{X}, X, leaky - ReLU, Dropout, GP, pac), \quad (5)$$

where \tilde{x} is the data produced from the generator and X is the input data used for the model. *Leaky-ReLU* is the activation function and *Dropout* is used for stability and over-fitting prevention. *GP* stands for Gradient Penalty [13] and is used in the loss term of the critic. *Pac* is a notion coming from PacGAN [31] and is used to prevent mode collapse in categorical data, which we found practically useful in terms of improving the outcomes.

3.3. Training

The final loss function takes the form of:

$$\begin{aligned} L_D &= \underbrace{\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)]}_{\text{Critic loss}} + \lambda \underbrace{\mathbb{E}_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D(\tilde{x}) - 1\|)^2]}_{\text{Gradient penalty}}, \\ L_G &= - \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})], \\ L_R &= L(x, x') + regularizer, \\ &\text{s.t. } \text{tr}[(I + \alpha A \circ A)^m] - m = 0, \end{aligned} \quad (6)$$

where $L(x, x')$ and *regularizer* are defined by Equation (3) and (4), respectively. In Equation (6) D is the discriminator and λ is the penalty coefficient used to calculate the gradient penalty associated with the Wasserstein metric. Distribution-wise, \mathbb{P}_g and \mathbb{P}_r are the generated and real distributions, while $\mathbb{P}_{\tilde{x}}$ is produced by sampling uniformly along a straight line between the aforementioned distributions.

We utilize the critic loss L_D to train the critic, the generator loss L_G to train the generator (namely the decoder in the AE), and the reconstruction loss L_R for both the encoder and decoder.

To enforce acyclicity, we use the acyclicity constraint proposed by Yu et al. [5], where A is the weighted adjacency matrix of the causal graph, m is the number of the variables, tr is a matrix trace and \circ is the Hadamard product [32] of A .

The acyclicity requirement associated with DAG structure learning reformulates the nature of the structure learning approach into a constrained continuous optimization. As such, we treat our approach as a constrained optimization problem and use the popular augmented Lagrangian method [33] to solve it.

In addition, our model naturally handles discrete variables by reformulating the reconstruction loss term using the Cross-Entropy Loss (CEL) as follows:

$$L(x, x') = - \sum_{i=1}^m \sum_{j=1}^d X_{ij} \log(P_X)_{ij} \quad (7)$$

where P_X is the output of the decoder and X is the input data for the auto-encoder.

4. EXPERIMENTS

This section provides experimental results of the model performance by comparing against other related approaches. In particular, our experiments try to identify the contribution from the Wasserstein loss to causal structure learning by making a direct comparison with DAG-GNN [5] where a similar auto-encoder architecture was used without involving the Wasserstein loss. Furthermore, we have also compared the results against DAG-NOTEARS [4] and DAG-NoCurl [29]. All the comparisons are measured using the Structural Hamming Distance (SHD) [34]. More specifically, we measure the SHD between the *learned causal graph* and the *ground truth graph*. Moreover, we also test the integrity of the generated data against CorGAN [7].

The implementation was based on PyTorch [35]. In addition, we used learning rate schedulers and Adam optimizers for both discriminator and auto-encoder with a learning rate of 3-e3.

4.1. Continuous data

To evaluate the model with continuous data, our experiments tried to learn causal graphs from synthetic data that were created with known causal graph structures and equations. To allow comparisons, we employed the same underlying graphs and equations like those in the related work, namely DAG-GNN [5], DAG-NoCurl [29] and DAG-NOTEARS [4].

More specifically, the data synthesis was performed in two steps: 1) generating the ground truth causal graph and 2) generating samples from the graph based on the linear SEM of the ground truth graph. In Step (1), we generated an Erdos-Renyi directed acyclic graph with an expected node degree of 3. The DAG was represented in a weighted adjacency matrix A . In Step (2), a

sample X was generated based on the following equations. We used the linear SEM $X = A^T x + z$ for the linear case, and two different equations, namely $X = A^T h(x) + z$ (non-linear-1) and $X = 2\sin(A^T(x+0.5)) + A^T \cos(x+0.5) + z$ (non-linear-2) for the nonlinear case. In particular, the two non-linear equations were selected because they were used in synthetic data experiments with similar models (DAG-GNN and all other models involved in its evaluation), which allows for more reliable model comparison and more comprehensive experiments.

The experiments were conducted with 5000 samples per graph. The graph sizes used in the experiments were 10, 20, 50 and 100. We measured the SHD (averaged over five different iterations of each model) between the output of a model and the ground truth, and the outcome was compared against those from the related work models (i.e. those mentioned at the beginning of Section 4). In addition to the mean SHD, confidence intervals were also measured based on the variance in the estimated means. These provide insight into the consistency of the model. Tables 1-3 show the results on continuous data samples:

Table 1. Comparisons of DAG Structure Learning Outcomes between DAG-NOTEARS, DAG-NoCurl, DAG-GNN and DAG-WGAN with Linear Data Samples

Model	SHD (5000 linear samples)			
	d = 10	d = 20	d = 50	d = 100
DAG-NOTEARS	8.4 ± 7.94	2.6 ± 1.84	25.2 ± 19.82	106.56 ± 56.51
DAG-NoCurl	7.9 ± 7.26	2.5 ± 1.93	24.6 ± 19.43	99.18 ± 55.27
DAG-GNN	6 ± 7.77	3.2 ± 1.6	21.4 ± 14.15	88.8 ± 47.63
DAG-WGAN	2.2 ± 4.4	2 ± 1.1	4.8 ± 4.26	28.20 ± 12.02

Table 2. Comparison of DAG Structure Learning Outcomes between DAG-NOTEARS, DAG-NoCurl, DAG-GNN and DAG-WGAN with Non-Linear Data Samples 1

Model	SHD (5000 non-linear-1 samples)			
	d = 10	d = 20	d = 50	d = 100
DAG-NOTEARS	11.2 ± 4.79	19.3 ± 3.14	53.7 ± 11.39	105.47 ± 13.51
DAG-NoCurl	10.4 ± 4.42	17.4 ± 3.27	51.6 ± 11.43	105.7 ± 13.65
DAG-GNN	9.40 ± 0.8	15 ± 3.58	49.8 ± 7.03	104.8 ± 12.84
DAG-WGAN	9.8 ± 2.4	16 ± 5.4	40.40 ± 10.97	80.40 ± 9.09

Table 3. Comparison of DAG Structure Learning Outcomes between DAG-NOTEARS, DAG-NoCurl, DAG-GNN and DAG-WGAN with Non-Linear Data Samples 2

Model	SHD (5000 non-linear-2 samples)			
	d = 10	d = 20	d = 50	d = 100
DAG-NOTEARS	9.8 ± 2.61	22.9 ± 2.14	38.3 ± 13.19	125.21 ± 61.19
DAG-NoCurl	7.4 ± 2.78	17.6 ± 2.25	33.6 ± 12.53	116.8 ± 62.3
DAG-GNN	2.6 ± 2.06	3.80 ± 1.94	13.8 ± 6.88	112.2 ± 59.05
DAG-WGAN	1 ± 1.1	3.4 ± 2.06	12.20 ± 7.81	20.20 ± 11.67

4.2. Benchmark discrete data

To evaluate the model with discrete data, we used the benchmark datasets available at the Bayesian Network Repository <https://www.bnlearn.com/bnrepository/>. The repository provides a variety of datasets together with their ground truth graphs (Discrete Bayesian Networks, Gaussian Bayesian Networks and Conditional Linear Gaussian Bayesian Networks) in different sizes (Small Networks, Medium Networks, Large Networks, Very Large Networks and Massive Networks). To test the scalability of our model, we used datasets of multiple sizes. The datasets utilized in the experiment were Sachs, Alarm, Child, Hailfinder and Pathfinder. The SHD metric was used to measure the performance. Table 4 contains the results from the experiment.

Table 4. Comparison of DAG Structure Learning Outcomes between DAG-WGAN and DAG-GNN with Discrete Data Samples

Dataset	Nodes	SHD	
		DAG-WGAN	DAG-GNN
Sachs	11	17	25
Child	20	20	30
Alarm	37	36	55
Hailfinder	56	73	71
Pathfinder	109	196	218

4.3. Data Generation

DAG-WGAN was also evaluated by comparing its data generation capabilities against other models. More specifically, we compare the data generation capabilities of the models on a 'dimension-wise probability' basis by measuring how well these models learn the real data distributions per dimension. We used the *MIMIC-III* dataset [36] in the experiments as the same dataset was also used in other comparable works. The data is presented in the form of a patient record, where each record has a fixed size of 1071 entries.

Figure 2 depicts the results of the experiment. We have only compared with CorGAN [7] as it out-performs the other similar models such as medGAN [37] and DBM [38] - see [7] where results of the other models are available. We present the results in a scatter plot, where each point

represents one of the 1071 entries and the x and y axes represent the success rate for real and synthetic data respectively. In addition, we use a diagonal line to mark the ideal scenario.

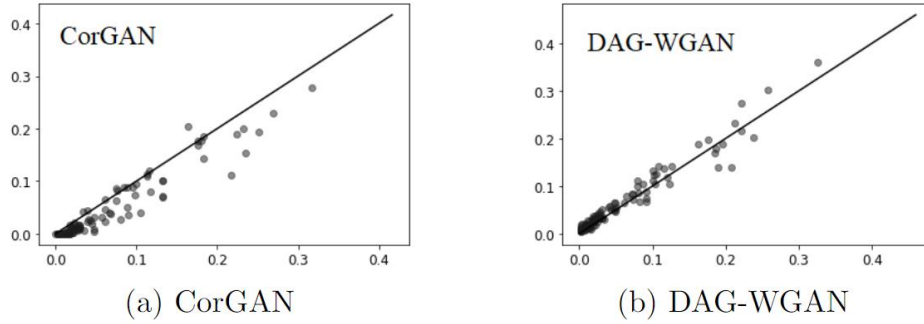


Figure 2. Data generation test results

5. DISCUSSION

The results on the continuous datasets are competitive across all three cases (linear, non-linear-1 and non-linear-2). According to Tables 1, 2 and 3, our model dominates DAG-NOTEARS and DAG-NoCurl, producing better results throughout all experiments and outperforming DAG-GNN in most of the cases.

For small-scale datasets (e.g. $d=10$ and $d=20$), our model performs better than DAG-GNN in most cases and is superior to DAG-NOTEARS and DAG-NoCurl in all the experiments.

For large-scale datasets (e.g. $d=50$ and $d=100$), our model outperforms all the other models used in the study by a significantly large margin, which implies that the model can scale better. This is a significant advantage.

Notably, our experiments have mainly focused on the comparisons with DAG-GNN, as we aim to identify contributions from the Wasserstein loss to causal structure learning. In DAG-GNN, a very similar auto-encoder architecture was employed and DAG-WGAN has added WGAN as an additional component. Hence the comparison is meaningful in order to identify contributions from the Wasserstein metric.

For the discrete case, the results from the comparison between the two models are competitive. Out of the five experiments conducted during the study, four results were clearly in favor of our model (i.e. DAG-WGAN) and in one case DAG-GNN was slightly better.

Also, according to the results illustrated in Figure 2, dimension-wise, the data generated using DAG-WGAN is more accurate and of higher quality than the ones generated using CorGAN, medGAN or DBM.

These results show that DAG-WGAN can handle both continuous and discrete data effectively. They have also demonstrated the quality of the generated data. As the improvement was achieved by introducing the Wasserstein loss in addition to the auto-encoder architecture, the comparisons between them show that the hypothesis on the contribution from the Wasserstein metric to causal structure learning stands.

However, the discrepancy which occurred in the synthetic continuous data results provides us with an insight into the limitations of the model. Some of our early analysis shows that further

improvement can be achieved by generalizing the current auto-encoder architecture. Furthermore, as it stands currently, our model does not handle vector or mixed-typed data. These aspects will be further experimented with and reported in our future work.

On the topic of potential improvements, the capability of recovering latent representation places the generative models in a good position to address the hidden confounder challenges in causality learning - some earlier work from [39-40] have moved towards this direction. We will further investigate whether DAG-WGAN can contribute. Last but not least, the latest work in DAG-NoCurl [29] shows that the speed performance can be improved by avoiding the DAG constraints. We will investigate how this new development can be adapted to DAG-WGAN to improve its overall performance.

6. CONCLUSION

This work studies the use of the Wasserstein distance metric for causal structure learning from tabular data. We investigate if the inclusion of the Wasserstein metric as an adversarial loss can simultaneously improve structure learning while generating more realistic data samples. This leads to a novel approach for learning causal structures, which we coined DAG-WGAN. The effectiveness of DAG-WGAN has been demonstrated through a series of experiments, which show that the new model using the Wasserstein metric can indeed improve the outcomes of causal structure learning. The improved quality of the synthesized data in turn leads to an improvement in causal structure learning.

REFERENCES

- [1] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Los Angeles, California (1988)
- [2] Chickering, D.M., Heckerman, D., Meek, C.: Large sample learning of Bayesian networks is np-hard. *Journal of Machine Learning Research* 5, 1287–1330 (2004)
- [3] Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* 65(1), 31–78 (2006)
- [4] Zheng, X., Aragam, B., Ravikumar, P., Xing, E.P.: DAGs with NOTEARS: Continuous Optimization for Structure Learning. Paper presented at 32nd Conference on Neural Information Processing Systems, Montréal, Canada (2018)
- [5] Yu, Y., Chen, J.J., Gao, T., Yu, M.: DAG-GNN: DAG Structure Learning with Graph Neural Networks. Paper presented at Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s) (2019)
- [6] Lachapelle, S., Brouillard, P., Deleu, T., Lacoste-Julien, S.: Gradient-Based Neural DAG Learning. Paper presented at ICLR 2020 (2020)
- [7] Torfi, A., Fox, E.A.: CORGAN: Correlation-Capturing Convolutional Neural Networks for Generating Synthetic Healthcare Records. Paper presented at The Thirty-Third International FLAIRS Conference (FLAIRS-33) (2020)
- [8] Gao, Y., Shen, L., Xia, S.-T.: DAG-GAN: Causal Structure Learning with Generative Adversarial Nets. Paper presented at IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2021)
- [9] Kantorovich, L.V.: Mathematical methods of organizing and planning production. *Management Science* 6(4), 366–422 (1960)
- [10] Cuturi, M.: Sinkhorn Distances: Lightspeed Computation of Optimal Transport. Paper presented at Advances in Neural Information Processing Systems 26 (2013)
- [11] Jacob, L., She, J., Almahairi, A., Rajeswar, S., Courville, A.C.: W2GAN: RECOVERING AN OPTIMAL TRANSPORT MAP WITH A GAN. Paper presented at ICLR 2018 (2018)
- [12] Genevay, A., Peyré, G., Cuturi, M.: Learning Generative Models with Sinkhorn Divergences. Paper presented at 21st International Conference on Artificial Intelligence and Statistics (2018)

- [13] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. Paper presented at Proceedings of the 34th International Conference on Machine Learning (2017)
- [14] Wiatrak, M., Albrecht, S.V., Nystrom, A.: Stabilizing Generative Adversarial Networks: A Survey. Preprint at <https://arxiv.org/abs/1910.00927> (2019)
- [15] Spirtes, P., Glymour, C.: An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review* 9(1), 62–72 (1991)
- [16] Pearl, J.: Causality: models, reasoning, and inference. *Econometric Theory* 19(46), 675–685 (2003)
- [17] Spirtes, P., Meek, C., Richardson, T.: Causal Inference in the Presence of Latent Variables and Selection Bias. Paper presented at UAI'95: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence (1995)
- [18] Zhang, J.: On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence* 172(16-17), 1873–1896 (2008)
- [19] Kuipers, J., Moffa, G., Heckerman, D.: Addendum on the scoring of Gaussian directed acyclic graphical models. *The Annals of Statistics* 42(4), 1689–1691 (2014)
- [20] Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20, 197–243 (1995)
- [21] Chickering, D.M., Heckerman, D.: Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning* 29(2-3), 181–212 (1997)
- [22] Bouckaert, R.: Probabilistic network construction using the minimum description length principle. Paper presented at European conference on symbolic and quantitative approaches to reasoning and uncertainty (1993)
- [23] Nie, S., Maua, D., de Campos, C., Ji, Q.: Advances in Learning Bayesian Networks of Bounded Treewidth. Paper presented at Advances in Neural Information Processing Systems 27 (2014)
- [24] Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory* 14(3), 462–467 (1968)
- [25] He, R., Tian, J., Wu, H.: Structure learning in Bayesian networks of a moderate size by efficient sampling. *Journal of Machine Learning Research* 17(1), 3483–3536 (2016)
- [26] Madigan, D., York, J., Allard, D.: Bayesian graphical models for discrete data. *International Statistical Review / Revue Internationale de Statistique* 63(2), 215–232 (1995)
- [27] Friedman, N., Koller, D.: Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* 50(1-2), 95–125 (2003)
- [28] Fast, A., Jensen, D.: Constraint Relaxation for Learning the Structure of Bayesian Networks. Paper presented at University of Massachusetts Amherst (2009)
- [29] Yue Yu, N.Y. Tian Gao, Ji, Q.: DAGs with No Curl: An Efficient DAG Structure Learning Approach. Paper presented at Proceedings of the 38th International Conference on Machine Learning (2021)
- [30] Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Auto-encoding beyond Pixels Using a Learned Similarity Metric. Paper presented at Proceedings of the 33rd International Conference on International Conference on Machine Learning (2016)
- [31] Cheng, A.: PAC-GAN: Packet Generation of Network Traffic using Generative Adversarial Networks. Paper presented at IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (2019)
- [32] Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge, New York (1985)
- [33] Bertsekas, D.: *Nonlinear Programming*. Athena Scientific, 2nd edition (1999)
- [34] de Jongh, M., Druzdzel, M.J.: A comparison of structural distance measures for causal Bayesian network models. *Recent Advances in Intelligent Information Systems*, 443–456 (2009)
- [35] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. Paper presented at 33rd Conference on Neural Information Processing Systems, Vancouver, Canada (2019)
- [36] Johnson, A.E.W., Pollard, T.J., Shen, L., Lehman, L.-w.H., Feng, M., Ghassemi, M.M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: MIMIC-III, a freely accessible critical care database. <https://doi.org/10.1038/sdata.2016.35> (2016)

- [37] Choi, E., Biswal, S., Malin, B.A., Duke, J.D., Stewart, W.F., Sun, J.: Generating Multi-Label Discrete Patient Records using Generative Adversarial Networks. Paper presented at Proceedings of Machine Learning for Healthcare 2017 (2017)
- [38] Salakhutdinov, R., Hinton, G.E.: Replicated Softmax: An Undirected Topic Model. Paper presented at Advances in Neural Information Processing Systems 22 (2009)
- [39] Louizos, C., Shalit, U., Mooij, J.M., Sontag, D.A., Zemel, R.S., Welling, M.: Causal Effect Inference with Deep Latent-Variable Models. Preprint at <https://arxiv.org/abs/1705.08821> (2017)
- [40] Wang, Y., Blei, D.M.: The blessings of multiple causes. *Journal of the American Statistical Association* 114(528), 1574–1596 (2019)

AUTHORS

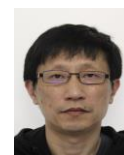
Hristo Petkov received the B.Sc. (First Class) degree from the Department of Computer and Information Sciences, University of Strathclyde. He is currently pursuing the Doctor's degree with the same department. His research interests include medicine, healthcare, deep learning, neural network models.



Colin Hanley received the M.Sc. degree from the Department of Management Science, University of Strathclyde. Currently, he is in pursuit of a career as a Full Stack Data Developer. His interests include quantitative analysis, human decision making, data analytics.



Feng Dong is a Professor of Computer Science at the University of Strathclyde. He was awarded a PhD in Zhejiang University, China. His recent research has addressed human centric AI to support knowledge discovery, visual data analytics, image analysis, pattern recognition.



APPLICATION OF CROSS-WAVELET AND SINGULAR VALUE DECOMPOSITION ON COVID-19 AND BIO-PHYSICAL DATA

Iftikhar U. Sikder¹ and James J. Ribero²

¹Department of Information Systems, Cleveland State University, USA

²IBA, University of Dhaka, Bangladesh

ABSTRACT

The paper examines the bivariate relationship between COVID-19 and temperature time series using Singular Value Decomposition (SVD) and continuous cross-wavelet analysis. The COVID-19 incidence data and the temperature data of the corresponding period were transformed using SVD into significant eigen-state vectors for each spatial unit. Wavelet transformation was performed to analyze and compare the frequency structure of the single and the bivariate time series. The result provides coherency measures in the ranges of time period for the corresponding spatial units. Additionally, wavelet power spectrum and paired wavelet coherence statistics and phase difference were estimated. The result suggests statistically significant coherency at various frequencies. It also indicates complex conjugate dynamic relationships in terms phases and phase differences.

KEYWORDS

COVID-19, SVD, Wavelet analysis, Cross-wavelet power, Wavelet coherence.

1. INTRODUCTION

The outbreak of COVID-19 has significantly changed the landscape of the global health. However, the dynamics between the bio-physical or climatic variables and the diffusion of COVID-19 is poorly understood [1]-[3]. There are various claims with regards to the dependency between the incidence or prevalence and environmental variables. It has often been argued that lower (cold) temperature act as a catalyst in significantly increasing the spread of COVID-19 [4]-[5]. There also exist alternative claims that warm temperatures slow down the spread of COVID-19 [6]. In contrast to these claims, some scholars assert that temperature does not play any role in the spread of COVID-19 [7]. In this paper, we have examined some specific empirical relationships of such dependencies, namely wavelet coherence and its statistical significance, phases and phase differences using the dataset of the USA.

2. OBJECTIVES AND SCOPE

The primary objective of the paper is to characterize the dynamic relationship of COVID-19 and a time-variant bio-physical parameter namely the temperature. This paper aims to provide an empirical investigation that captures and analyzes the characteristic relationship of these variables. The study area was limited within the United States. The data for COVID-19 cases was collected from the fifty (50) states, and the corresponding data on temperature of the same period was collected from these states. The period covered was between Jan. 21, 2020, till date. Around

40,000 records (20000 COVID-19 data records and 20000 temporal temperature data records) have been collected and used for the research [8]-[9].

3. METHODOLOGY

The variables used in the model are featured as time series data, and thus expected to fluctuate with an associated noise. Employing conventional smoothing technique involving amplitude-based statistical analysis would not be appropriate to achieve the research objective. Therefore, we adopt a Wavelet Transform algorithm not only to capture the periodicities of the variables over the time, but also to establish coherence among the variables in the frequency domain.

3.1. Modeling Framework

The Figure 1 below details the process flow of the research:

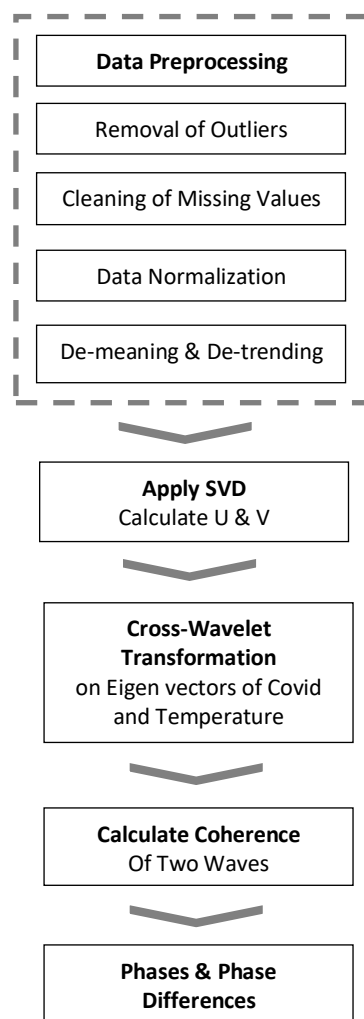


Figure 1. Methodology flow chart

3.1.1. Data Pre-Processing

The data sets on COVID-19 cases and temperature across the 50 states accessed in a format that

was not readily available for analysis. The pre-processing steps involved the basic data cleaning functions such as removal of irrelevant attributes and missing values, removal of outliers, de-meaning, linear detrending, and data normalization. The data processing was done in R environment using *WaveletComp* package [10]. The outputs of the pre-processing are data frames that were transformed into rectangular matrices, where the rows represent either COVID-19 cases by states or temperature, and the columns represent the date.

3.1.2. Calculating Singular Value Decomposition (SVD)

A widely adopted matrix factorization or dimension reduction technique namely Singular Value Decomposition (SVD) was applied on both the COVID-19 and temperature data sets to compress the data into ortho normal eigen basis to rectangular matrices. Based on the top singular values, top three eigen states for both COVID-19 and temperature were selected, which in combination accounts for significant total variance of the original data. Plots of the transformed data set are displayed in Figure 2 below showing negative correlation (with *correlation coefficient* of -0.34) of cases and temperature.

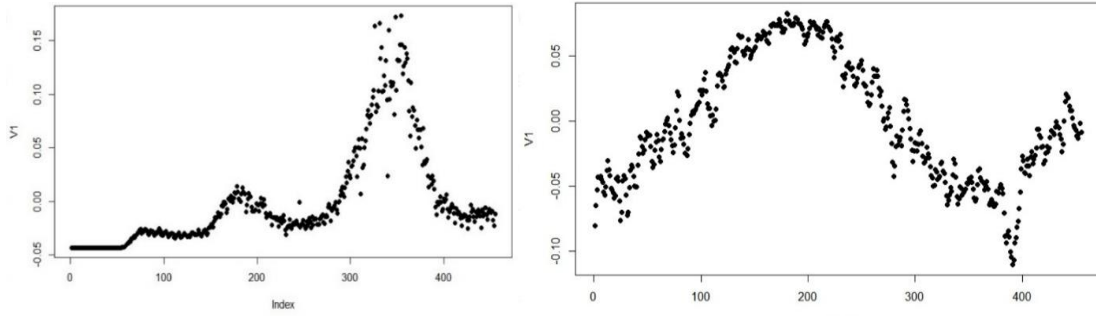


Figure 2. Significant eigenvectors of temperature and Covid-19 cases

3.1.3. Calculating Wavelet Transformation

A Wavelet transform decomposes a time series into a set of wavelets localized in time. The Wavelet transformation was performed on both COVID-19 and temperature time-series of *eigen* vectors. We applied *Morlet* continuous Wavelet transforms to the transformed data using the *WaveletComp* R package. Wavelet transformation leads to a continuous, complex-valued output of the time series that preserves both time and frequency resolution parameters. The transform is separable into its real part and imaginary part providing information on both local amplitude and instantaneous phase. This allows for the investigation of coherency between the two time series. Given two time series $X(t)$ and $Y(t)$, and corresponding *wavelet spectrums* $W_x(s, \tau)$ and $W_y(s, \tau)$ which could be considered as localized energy spectrum varying with scale s , and translation τ , and associated frequency ω and time t . The cross-wavelet transformation $W_{xy}(s, \tau)$ is associated with complex-valued wavelet coherency:

$$Y(s, \tau) = \frac{\langle W_{xy}(s, \tau) \rangle}{\sqrt{\langle W_x(s, \tau) \rangle \langle W_y(s, \tau) \rangle}} \quad (1)$$

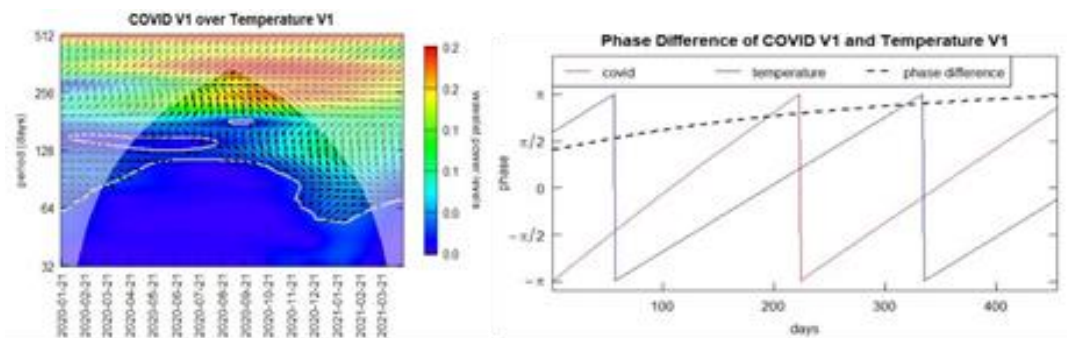
and the normalizing wavelet power spectra coherence is:

$$\gamma^2(s, \tau) = \frac{[\langle \text{Re}(W_{xy}(s, \tau)) \rangle]^2 + [\langle \text{Im}(W_{xy}(s, \tau)) \rangle]^2}{\langle W_x(s, \tau) \rangle \langle W_y(s, \tau) \rangle} \quad (2)$$

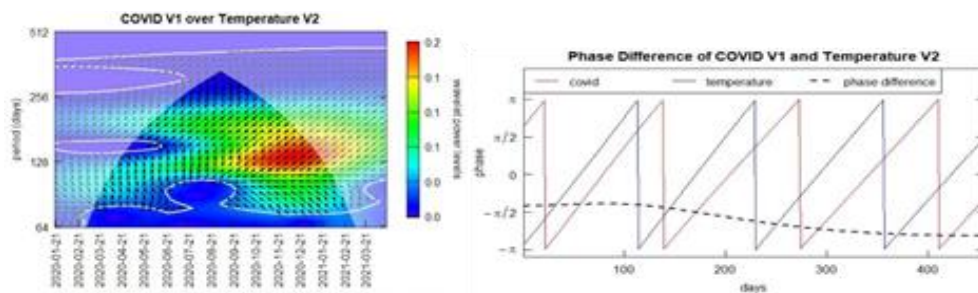
The angle brackets $\langle \rangle$ denotes the smoothing operation over time. The squaring of the amplitude component gives us the wavelet power spectrum $0 \leq Y^2(s, \tau) \leq 1$, which is some what analogous to conventional correlation coefficient. After computing the wavelet power spectrum for each of *eigenvector*, we analyze the coherence of the paired waves of COVID-19 and temperature using the coherence function. The phase lags between the variables were also computed. The cross-wavelet transformation provided cross-magnitude, phase differences, non-stationarity, and coherency between signals. Using these results of the cross-wavelet transformation, a series' synchronicity at certain periods and across certain ranges of time was analyzed.

4. RESULTS AND MODEL INTERPRETATION

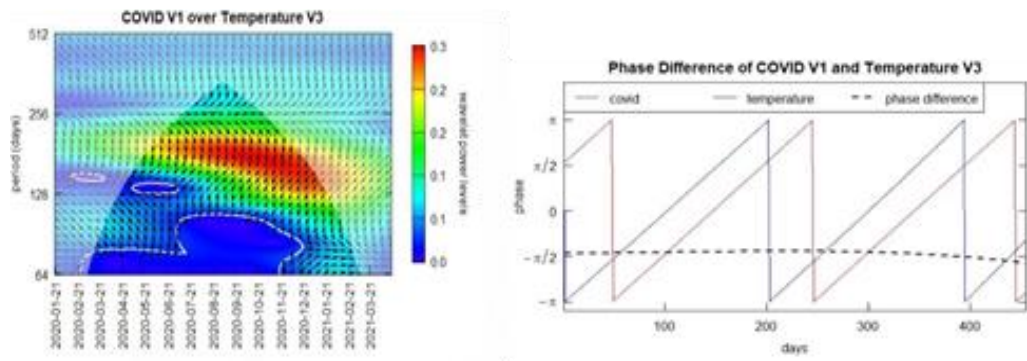
The cross-wavelet analysis generated coherence plot that shows that that there is a coherence (correlation) between COVID-19 and temperature, and these relationships are statistically significant (the region enveloped or bounded by the white line). The phases and phase differences show varied results. Figures 3a, 3b, 3c shows COVID-19 and temperature are out of phase with varying phase lags while Figure 3d and 3e shows that are in phase. Comparing the result of plots 3a and 3e, COVID-19 and temperature were both out of phase in 3a, with temperature leading and COVID-19 lagging by 96days, while from 3e both time series were in phase. Though temperature was leading, the lag period was much narrow (around 5 days) compared to 3a.



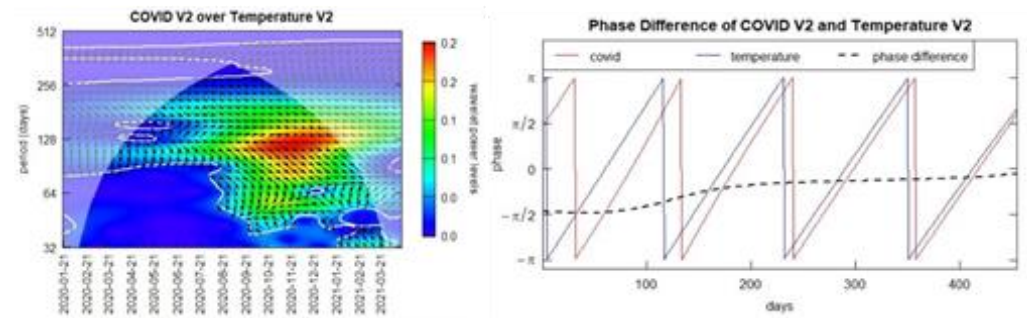
3a. COVID-19 V1 & Temperature V1



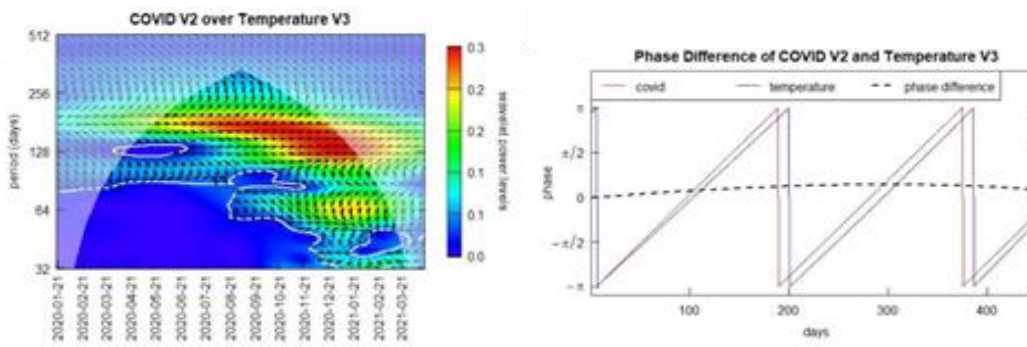
3b. COVID-19 V1 & Temperature V2



3c. COVID-19 V1 & Temperature V3



3d. COVID-19 V2 & Temperature V2



3e. COVID-19 V2 & Temperature V3

Figure 3. Cross-wavelet analysis generated coherence plots

5. CONCLUSION

This article studies the dynamics between two-time series variables – COVID-19 and temperature – using SVD and wavelet transform approach. The results from the continuous cross-wavelet transform shows power spectrum strengths and coherence corresponding at various frequencies (periods). The coherence statistics suggest statistically significant relationship. The results also show varying phases and phase lags with leading and lagging behavior showing complex conjugate dynamics. Future studies focusing on spatially explicit mapping of coherence could provide additional explanatory schemes and better understanding of the spatio-temporal dynamics of the disease.

REFERENCES

- [1] Sahoo, P.K., Powell, M.A., Mittal, S., and Garg, V. (2020) “Is the transmission of novel coronavirus disease (COVID-19) weather dependent?”, *Journal of the Air & Waste Management Association*, , Vol. 70, (11), pp. 1061-1064
- [2] Kroumpouzos, G., Gupta, M., Jafferany, M., Lotti, T., Sadoughifar, R., Sitkowska, Z., and Goldust, M. (2020) “COVID-19: A relationship to climate and environmental conditions?”, *Dermatologic therapy*
- [3] Rosario, D.K., Mutz, Y.S., Bernardes, P.C., and Conte-Junior, C.A. (2020) “Relationship between COVID-19 and weather: Case study in a tropical country”, *International journal of hygiene and environmental health*, Vol. 229, pp. 113587
- [4] Wang, M., Jiang, A., Gong, L., Luo, L., Guo, W., Li, C., Zheng, J., Li, C., Yang, B., and Zeng, J. (2020) “Temperature significant change COVID-19 Transmission in 429 cities”, *MedRxiv*
- [5] Prata, D.N., Rodrigues, W., and Bermejo, P.H. (2020) “Temperature significantly changes COVID-19 transmission in (sub) tropical cities of Brazil”, *Science of the Total Environment*, , 729, pp. 138862
- [6] Oliveiros, B., Caramelo, L., Ferreira, N.C., and Caramelo, F. (2020) “Role of temperature and humidity in the modulation of the doubling time of COVID-19 cases”, *MedRxiv*
- [7] Jamil, T., Alam, I., Gojobori, T., and Duarte, C.M. (2020) “No evidence for temperature-dependence of the COVID-19 epidemic”, *Frontiers in public health*, , 8, pp. 436
- [8] COVID-19 Dataset. <https://www.kaggle.com/joelhanson/coronavirus-COVID-19-data-in-the-united-states?select=us-states.csv> [Accessed on 1st September 2021]
- [9] Temporal Temperature Dataset. <https://catalog.data.gov/dataset> [Accessed on 1st September 2021]
- [10] Roesch, A., Schmidbauer, H., and Roesch, M.A. (2014) “Package ‘WaveletComp’”, *The Comprehensive R Archive Network*

AUTHORS

Iftikhar U. Sikder is an Associate Professor jointly appointed in the department of Information Systems and the Electrical Engineering and Computer Science at Cleveland State University, USA. He holds a PhD in Computer Information Systems from the University of Maryland, Baltimore. His research interests include soft computing, granular computing, spatial databases, spatio-temporal data mining and adaptive complex systems. His papers appeared in the journal of Knowledge-Based Systems, Risk Analysis, Expert Systems with Applications, Intl. journal of Digital Earth, International Journal of Mobile Communications, and Information Resources Management Journal, Intl. Journal of Management & Decision Making, and Intl. Journal of Aerospace Survey and Earth Sciences and many others journals. He has authored many book chapters and presented papers in many national and international conferences. Dr. Sikder is also currently serving in the editorial board of International Journal of Computational Models and Algorithms in Medicine and Intl. Journal of Computers in Clinical Practice.



James J. Ribero is an Adjunct Faculty at IBA, Dhaka University, Bangladesh. He holds MBBS, MS (Microbiology) and MBA from Dhaka University, Bangladesh. His research interests include applications of Machine Learning and Big Data technologies into medical and life sciences. He has authored book chapters, monographs, and presented papers in many national and international conferences. He is an ex-executive editor of *The Orion* medical journal.



ARTIFICIAL INTELLIGENCE: FRAMEWORK OF DRIVING TRIGGERS TO PAST, PRESENT AND FUTURE APPLICATIONS AND INFLUENCERS OF INDUSTRY SECTOR ADOPTION

Richard Fulton¹, Diane Fulton² and Susan Kaplan³

¹Department of Computer Science, Troy University, Troy, Alabama, USA

²Department of Management,

Clayton State University, Morrow, Georgia, USA

³Modal Technology, Minneapolis, Minnesota, USA

ABSTRACT

To gain a sense of the development of Artificial Intelligence (AI), this research analyzes what has been done in the past, presently in the last decade and what is predicted for the next several decades. The paper will highlight the biggest changes in AI and give examples of how these technologies are applied in several key industry sectors along with influencers that can affect adoption speed. Lastly, the research examines the driving triggers such as cost, speed, accuracy, diversity/inclusion and interdisciplinary research/collaboration that propel AI into an essential transformative technology.

KEYWORDS

Artificial Intelligence, Key Industrial Sectors, Adoption of Technology, Driving Triggers, Technology Trends.

1. INTRODUCTION

Artificial Intelligence (AI) is an evolving science and art. Developments come in flashes and spurts over time. The scientific community changes its focus on different topics and applications. Technological developments can and will continue to expand the problem solving and innovative capabilities of AI. Researchers build on what has been done in the past, implement in the present and dream about what can happen in the future. Together, these developments over time lead to the state of the art of a technology like AI.

This paper presents a time-evolving Framework for AI (FAI) based on past and present adoptions and future expectations of technology uses. Triggers such as cost, speed, accuracy, customization, inclusivity/ diversity and cross discipline/collaboration are factors that push an organization to adopt and transform a new technology. When there are dramatic changes in the environment, what the customer needs, competitiveness in the industry and increased resources to implement a new technology, these become influencers in how rapidly technology becomes transformational as well. In this framework, the state of the art of AI is impacted by triggers, influencers and time. Three distinct industrial sectors including agriculture, education and healthcare illustrate the sector-dependent nature of AI application development over time,

spanning the past, present and future. The authors conclude with an in-depth discussion of the six driving triggers of AI transformative technology adoption.

2. RELATED WORKS

This section briefly reports the most related work to examining the triggers and influencers of AI technology adoption over time based upon a variety of theories and research models. The first group of theories and models pertinent to the development of artificial intelligence include those related to technology acceptance and adoption. The 3 most used acceptance/adoption models are the Technology Acceptance Model (TAM), Diffusion of Innovations Theory (DOI) and the Unified Theory of Acceptance and Use of Technology (UTAUT) [1].

TAM, the most widely tested empirical model, proposes three technology acceptance factors including 1) “perceived usefulness”, 2) “perceived ease of use” and 3) “attitude towards use” and focuses on the individual [2]. In contrast, DOI focuses on both individuals and organizations and four factors of time, channels of communication, social systems and innovation which impact technology diffusion and adoption [3]. Differences in adopter characteristics in the DOI model categorize firms and individuals within firms as early adopters, innovators, laggards, late majority and early majority leading credibility to industry sector differences discussed in this paper [3].

Lastly the UTAUT model is a compilation model built on 8 models (including TAM and DOI) emphasizing effort expectancy, performance expectancy, social influence, and facilitating conditions [4]. “Facilitating conditions” mean removal of barriers impeding technology adoption. For example, using the UTAUT model in applying new technology in e-learning, “facilitating conditions” included providing financial resources, new infrastructure, additional human resources and innovative educational content [5]. These “facilitating conditions” are included in this study under the “influencer” construct subcategory *resources*.

Both TAM and DOI models use the constructs of “perceived usefulness” and “relative advantage” [6]. A new construct of “perceived benefits of technology adoption” incorporated into the International Technology Adoption (ITA) model combined the previous constructs of technology utility to the individual with benefits to the company’s well-being and corresponds closely to the “triggers” of *speed*, *accuracy*, *cost*, and *customization* and the “influencers” of *competitive advantage* and *customer needs* presented in this paper.

Digital transformation in industry is a compelling topic and focus of a framework called “The Digital Transformation Journey” [7]. In their framework, the compelling construct of “mounting challenges and drivers” means finding ways to use technology to do business in new and better ways [7]. Coronavirus (Covid-19) in late 2019, for example, is considered a pressure point or “driver” of technology transformation in a variety of industrial sectors [7].

An example in the healthcare sector of a “driver” or “influencer” of AI technology is the coronavirus in Wuhan, China in 2019 which used AI tools to provide early detection of the coronavirus, isolating those areas with the virus [8]. It is likely the experience of a global pandemic will have a long-lasting and global impact on AI diffusion finding new ways of early detection which will help prevent future pandemics and influence health policies worldwide [8]. Increased competition is another “challenge” creating market pressure that if not addressed, can lead to loss of market share and revenues [7]. The Framework of AI uses the construct of “influencers” of *changing environments* and *competitive advantage* which correspond to transformational “challenges and drivers”.

In addition to building on previous model constructs of “benefits” such as this research’s “triggers” which add value, usefulness and benefits to the individual (less time to do a task) and organization (reducing costs and mistakes) and “influencers” of *changing environments*, *resources* and *competitive advantage*, the authors enhance the existing theories and models by adding two new “triggers of AI technology adoption” – *diversity/inclusion* and *cross-discipline/collaboration* in their framework. Not addressing these essential issues could sabotage transformational adoption of AI.

In fact, the more leaders understand the biases in technology [8] and the need for collaboration across disciplines/fields, the better they can improve its usefulness and therefore, increase its transformational adoption [9]. Lastly, this research supports the idea that AI technology is a dynamic phenomenon which changes over time and a better understanding of technology changes through past, present and future developments can help increase individual and organizational ability to build their AI maturity [10].

3. ARTIFICIAL INTELLIGENCE FRAMEWORK

Successes in one industry spur interest in another sector. Some sectors are quick to adopt new technological applications such as AI and others are more cautious. Factors that can prompt or influence adoption include changing environments such as climate change or a pandemic event like the 2019 coronavirus (covid-19) or evolving customer needs for a product or service [11][12][13].

Often, organizations are searching for competitive advantage such as cost, quality, or better satisfying a particular niche of consumers. For example, a recent McKinsey study showed advanced AI adopter firms were 52% more likely to increase their market share by 52% and 27% had growth in their marketplace compared to those who were testing or moderately implementing AI [14]. Lastly, there are changing priorities in the allocation and budgeting of resources depending on societal expectations and organizational readiness [15]. Figure 1 gives a schematic of the Framework for AI (FAI), from development triggers to adoption influencers based on past, present and future AI technology trends.

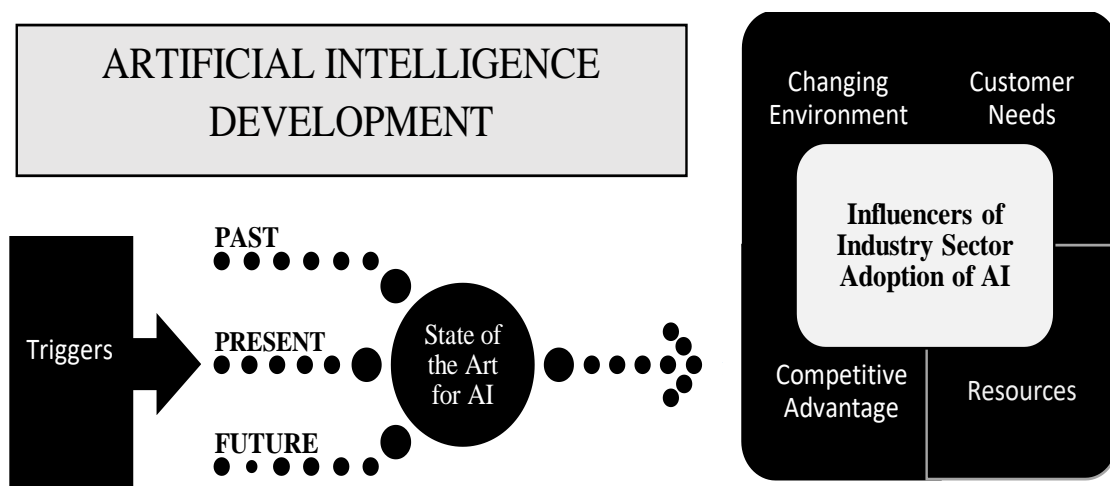


Figure 1. Framework for Artificial Intelligence: Triggers to Past, Present and Future Application Development and Influencers of Industry Sector Adoption

3.1. Past

AI began in the 1940s, demonstrating that a new form of computing was possible, with an approach derived from known cognitive processes and neurobiology. The initial purpose of AI was to automate, through computers, non-analytical human knowledge, from symbolic computation processes, connectionist ones, or a combination of both. AI was initially considered a branch of computer science with limited application and restricted by the capabilities of the hardware of the time.

Turing, a British mathematician, developed a code breaking computer called the *Bombe* in the early 1940's that successfully broke the Enigma code used by the Germans during World War II, a task thought impossible by most human mathematicians at the time. He also developed the Turing Test, that states "if a human is interacting with another human and a machine and unable to distinguish the machine from the human, then the machine is said to be intelligent" [16].

In 1956, John McCarthy offered one of the first and most influential definitions of AI: "The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it" [17].

One of the most famous AI examples is IBM's Deep Blue chess playing program, which beat the world chess champion Gary Kasparov in 1997. This expert system processed 200 million possible moves per second and determined the optimal next move looking 20 moves ahead [18].

3.2. Present

The current definition of artificial intelligence (AI) has transformed into "computing systems that are able to engage in human-like processes such as learning, adapting, synthesizing, self-correction and use of data for complex processing tasks" [19].

AI has become a vital element for the development of many services and industrial sectors in the 21st century. This discipline of computer science studies algorithms to develop computer solutions that copy the cognitive, physiological, or evolutionary phenomena of nature and human beings. The data, examples of solutions, or relationships between these facilitate the resolution of diverse problems [20]. AI exhibits, in certain aspects, "an intelligent behavior" that can be confused with that of a human expert in the development of certain tasks [21].

The Deep Blue project inspired the development of Watson, a computer that was able to beat the two best Jeopardy Game players in the world in 2011. Its software could process and reason using natural language, and draw from a massive supply of information poured into it in the months before the competition [22].

At present, AI has been redirected towards the construction of solutions to problems analyzing large volumes of data which change over time. Currently, the systems for approaching functions using iterative techniques, and the neural network architectures interconnected with each other, make up most of the techniques, which are grouped under the terms "Machine Learning" and "Deep Learning".

AI is becoming a growing presence in our society. From the intelligent sensors that make a car drive autonomously to mobile assistants, we are already surrounded by AI in some way or the other at all times [23]. Alexa, Siri, Cortana, security surveillance, fitness/dieting apps and online

customer service are all examples of AI [24]. A large portion of the global population use these products/services in their everyday lives and the demand and popularity are ever growing [24].

3.3. Future

AI is a game changing technology and disruptor. Within 10 years, it is predicted 375 million workers will need to change occupations as a result of widespread use of AI [24]. AI and machine learning are predicted to reshape most sectors but particularly manufacturing, energy, transportation, agriculture, labor markets, and financial management [25].

AI will not only impact our personal lives but also fundamentally transform how organizations make decisions and interact with employees and customers. One of the most vital questions will be how AI systems and humans can coexist with each other. Which decisions should be made by AI, which ones by humans, and which ones in collaboration will be an issues all companies need to address in the future [22].

4. KEY SECTOR APPLICATIONS: AGRICULTURE

4.1. Past

Agriculture is a sector that includes studies in science, engineering, and economics. The deductive techniques of AI expert systems have been used in the field of agriculture to integrate crop management which encompassed irrigation, nutritional problems and fertilization, weed control-cultivation, herbicide application, and insect control/insecticide application. Additional subject areas were plant pathology, salinity management, crop breeding, animal pathology, and animal herd management [26].

Agricultural applications of expert systems and decision support systems have also benefited the simulation of processes and the management of supply operations [27][28].

In other studies, AI has been used in quality control processes, whether or not they are supported by artificial vision [29] or in processes of justification of food policy decisions, such as when the use of AI is analyzed as a collaborative tool between the different actors that supply the agri-food chain, using distributed computing processes [30].

In the field of science, climate aspects are studied through modeling and solar radiation is predicted using neural networks [31][32].

4.2. Present

Interest in the application of AI to the world of agriculture and its multiple facets has been growing in recent years as it has proven to be a powerful tool for data analysis [33].

Current AI technology investigates the price behavior of agri-food products [34][35][36]. In these cases, artificial neural networks and machine learning techniques are applied to investigate the price variations of agricultural commodities.

The expansion and intensification of industrial and technological agriculture have increased production, decreased the number of people suffering from poor nutrition and ensured richer and more resource-intense diets around the world. Industrial agricultural activities also generate employment, improve economic growth and boost the service sector in industrial regions [37].

Agriculture 3.0 brought robotics and automation to the agricultural world, as evidenced by agricultural machinery that performs complete cycles of agricultural work such as planting, spraying, and harvesting [38][39][40][41].

4.3. Future

Now, agriculture 4.0, combines intelligent farms and the interconnection of machines and systems, and seeks to adapt production ecosystems by optimizing the use of resources such as water, fertilizers, and phytosanitary products. In addition, it uses big data and imaging technology to arrive at “precision agriculture” [42][43][44][45].

Combined with genetic engineering and the use of data, it can solve an important part of agriculture by maximizing efficiency in the use of resources and adapting to climate change and other challenges [46]. To this end, the use of big data in decision-making is essential [47][48]. The technification of agriculture, decision support systems and the inclusion of concepts of Industry 4.0 by agri-food companies will continue to generate increased innovation in AI [49].

5. APPLICATIONS IN EDUCATION SECTOR

5.1. Past

The IBM supercomputer Watson was watched across school and university campuses and all were delighted with the computer besting the 1994 world chess champion. In 2011, Watson with its victory in the game show *Jeopardy* against the two highest winners, heralded the era of cognitive computing with its potent natural language processing, knowledge representation and reasoning capabilities.

The educational interest in AI was initially captured through computers playing games but early versions of educational tutorials, learning management systems, simulations and iterative computer learning in the 1900s and early 2000s started the AI revolution in education [50][51][52].

5.2. Present

Universities have been particularly impacted by the 2019 coronavirus pandemic due to the in-person nature of traditional education. They are responding to this threat by investing in digital technologies such as cloud, AI, analytics, immersive learning spaces, and digital curricula. In fact, more than 80% of institutions are allocating over 25% of their 2021 IT budgets toward digital initiatives [53].

“Customization of learning has been happening through rising numbers of adaptive education programs, gaming, and software. These systems are personalized by enabling repeated lessons that students haven't mastered, and generally helping students to work at their own pace, space and liberty” [23].

Individualized automated tutoring has been developed to help students to learn easily and on their own schedules [54]. At Colorado State University, online students and tutors are using AI powered by Cognii, an Edtech company, to improve learning and assessment tools [55].

Another recent example of AI advancement is AlphaGo—a software or ‘machine learning’ developed by DeepMind, the AI branch of Google—that was able to defeat the world’s best

player at Go, a very complex board game considered more difficult than chess [56]. The AlphaGo program proved that the computer and deep learning can reach new heights and further advance human understanding in certain topics.

‘Machine learning’ is a subfield of artificial intelligence that includes software able to recognize patterns, make predictions, and apply the newly discovered patterns to situations that were not included or covered by their initial design.

5.3. Future

AI has the potential to modify the quality, quantity, delivery, and nature of education. It also promises to change forever the role of parents, students, teachers, and educational systems. Using Artificial Intelligence systems, software and support, students can learn from across the world at any time. These kinds of applications are taking the place of certain types of classroom instruction and may replace teachers in some cases [23].

AI can contribute to changing education via the automation of administrative teaching tasks, software programs that favor personalized education, the detection of topics that need reinforcement in class, the guidance and support of students outside the classroom, and the use of data in an intelligent way to teach and support the students [57].

Three techniques of AI are particularly relevant for future educational developments – personalization systems (knowledge and individualized adaptation of the student), software agents (intelligent programs and robots with autonomy and the ability to learn) [58] and ontologies and semantic web [59].

When developed and applied in education, these systems and techniques can be powerful resources for improving the teaching–learning process, since they are able to generate a kind of virtual teacher who is fully trained and has human characteristics, yet is able to interact ubiquitously (that is, at any time and place) [54].

By harnessing the power of AI and deep learning, educators can gain insights from the vast quantities of data collected from their students, make better decisions and improve student retention. Teachers can access detailed feedback on how learners are processing information. Big data can help answer key online learning questions—what are the most ideal ways to teach complex ideas and which parts of a course are best taught in person instead of online. Big data helps students find the right courses; customize them to their needs and keep them on the right track [55].

Most EdTech products will have an AI or deep learning component in the future. AI could help online learners self-assess, increase connectivity in global classrooms and create social simulation. Limitations include the uncertainty of how humans learn and fears among faculty that they must be retrained or could be displaced completely [55].

"Remote learning will coexist with on-campus education. As institutions accelerate their focus on student diversity and address unique educational needs, it is critical for them to make necessary technological investments to support their teaching models" [53].

In the future, higher educational institutions should expand outreach by using online courses and digitization of content to enable on-demand access by students across different geographies for remote learning, self-directed learning or specialized skill development. Secondly, increase funding to facilitate online learning, particularly enhancing IT capabilities – cloud platforms,

collaborative tools, data security measures, AI bots and assessments. Lastly, educational organizations must learn to mine data assets and use AI's analytical solutions to develop personalized content, upskill faculty and enable remote proctoring, communications and virtual assistants [53].

6. APPLICATIONS IN MEDICAL/HEALTHCARE SECTOR

6.1. Past

A recent review of the history of clinical decision support states the dramatic improvement in the medical sector due to the advent of cognitive aids to support diagnosis, treatment, care-coordination, surveillance and prevention, and health maintenance or wellness [60][61].

Some studies highlighted the importance of AI in healthcare, especially in medical informatics but there is still work to be done on examining the impacts and consequences of the technology [61][62].

6.2. Present

In the medical profession, image recognition tools are already outperforming physicians in the detection of skin cancer [63]. Molecular imaging modalities have also been effective in diagnosing neurodegenerative diseases [64].

Digital medicine and wearable devices are presently used in healthcare by mining data for anomaly detection, prediction, and diagnosis/decision making. Wearable devices and sensors have been used to continuously track physiologic parameters which guided patient care strategy that improved outcomes and lowered healthcare costs in cardiac patients with heart failure [65]. They also have been effective to improve diagnosis and management in neurological disorders such as Parkinson's disease [66].

Machine learning applications in healthcare have been helpful in earlier disease detection and prediction. For example, machine learning models were used in identifying stable subsets of predictive features for autism behavioral detection and blood biomarkers for autism [67][68].

Machine-learning algorithms were also used in the prediction of periventricular leukomalacia in neonates after cardiac surgery [69].

6.3. Future

Deep learning for automated and/or augmented biomedical image interpretation will continue to be used in radiology, pathology, dermatology, ophthalmology and cardiology with strict protocols and benchmarks in place to ensure data integrity and fairness. However, sensor-based, quantitative, objective and easy-to-use systems for assessing many diseases has the potential to replace traditional qualitative and subjective ratings by human interpretation in the future [70].

Future AI in healthcare must be able to use machine learning to handle structured data such as images, data, genetic data, and natural language processing to mine unstructured texts. Then it must be trained through healthcare data before it can assist physicians with disease diagnosis and treatment options [71].

AI in medicine will continue with informatics approaches from deep learning information management to control of health management systems, including electronic health records, and active guidance of physicians in their treatment decisions. Also in the future, healthcare will increase its use of robots to assist elderly patients and targeted nanorobots, a unique new drug delivery system [72].

7. TRIGGERS

Certain factors are accelerating the growth and use of AI throughout our society and will continue to be triggers for AI's transformative impact. AI can be used as a competitive strategy in all economic sectors particularly in cost/pricing advantages, customizing or personalizing products and services, and research using data mined from present and potential customers.

In addition, many AI advances have been accomplished by finding ways to increase the speed and accuracy of data resources and data research which can accelerate innovations while increasing the level of quality for consumers. Lastly, in our pursuit of the positive contributions of AI, we must be mindful of creating products and services that appeal to an inclusive and diverse group of people. Another way to increase the potential of AI is to use collaboration and reach across disciplines and sectors. Please see Figure 2 for the critical triggers impacting AI.

7.1. Speed

Artificial intelligence systems can take control of many factors in an organization. For example, in an educational classroom – AI can control time-consuming tasks like accounting processes, record keeping, filling out forms, producing documents and automatically grade assignments freeing up time for teachers to improve the quality of learning, increase active learning and help students when needed [73].

In a survey about the benefits of AI in the workforce, 61% of respondents said it helped them have a more efficient and productive workday [74]. Almost half (49%) felt it improved their decision-making and accelerated time to insights, while 51% said they believed AI enabled them to achieve a better work/life balance [75].

The three highest rated tasks to benefit from AI adoption were: 1) understanding trends and patterns; 2) moving data from one place to another and 3) accessing data residing in different places across the organization [74].

7.2. Accuracy

It is also predicted that 70% or more of companies will use some type of Artificial Intelligence in their operations because AI builds efficiency and effectiveness [24].

In the healthcare field, for example, AI can use sophisticated algorithms to 'learn' features from healthcare data, which can bring about insights for clinical practice and because it can be equipped with learning and self-correcting abilities, will improve its accuracy based on feedback over time [71].

In a recent interview with Susan Kaplan, the VP of a high-tech firm called Modal Technology located in Minneapolis, she cited that the "joint venture partnership between Modal Technology and medical researchers and scientists at McGill University Health Center Research in Montreal, Canada, using a new and mathematically proven non-statistical AI training model, ALIX,

increased accuracy of finding patients who had cancer”. Also, a “biprodut of the training identified and rank ordered the biomarkers from the most relevant to irrelevant. The glass box solution was explainable and repeatable”. Such abilities will help in “early detection of cancers, increase precision medicine solutions for patients and treatment outcomes in the future” [78].

7.3. Cost

Three cost-saving AI solutions include virtual assist (chatbots), human assist (which routes complex customer questions to a human), and screen assist (which provides common answers to humans) [79].

These AI technologies can save millions of dollars for financially stressed businesses in today’s challenging times by enabling them to address issues that affect customer service, costs and revenues [79].

AI has already increased productivity and efficiency in healthcare delivery, which has helped improve care outcomes, patient experiences and access to medical services [80].

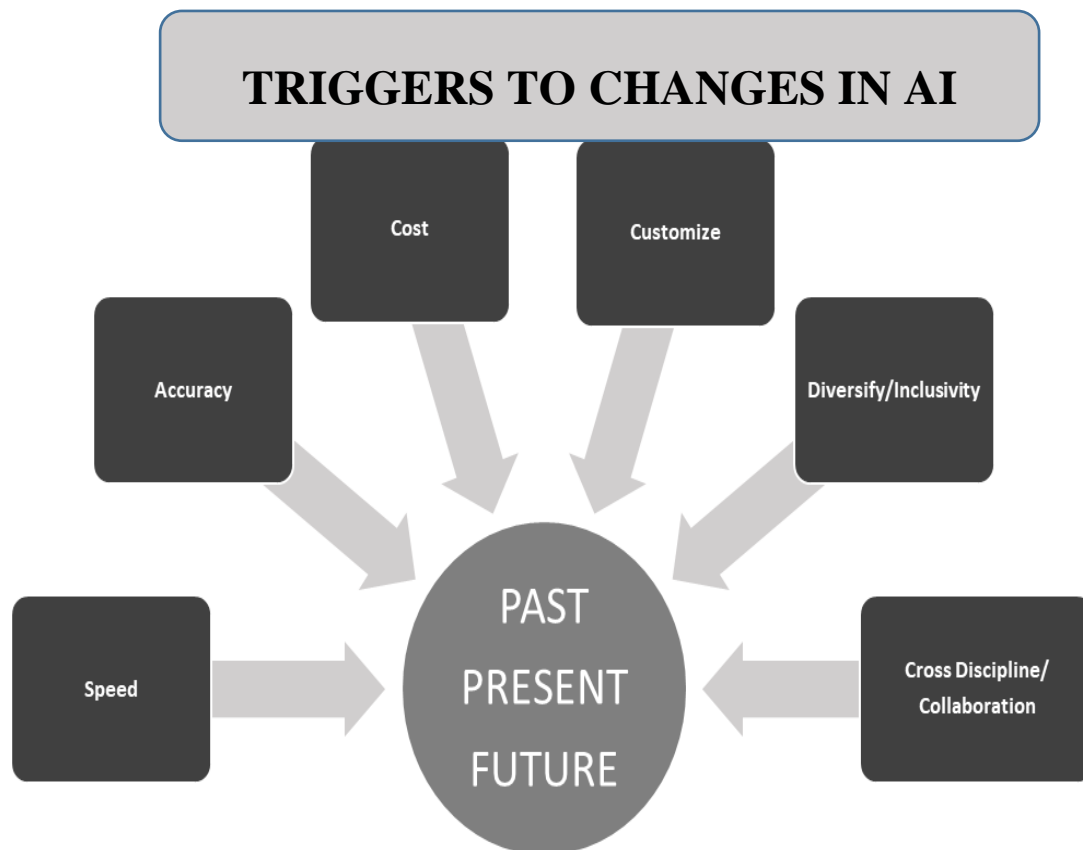


Figure 2: Triggers to Changes in Artificial Intelligence: Past, Present and Future

7.4. Customization

Customization is the name of the game – industries are using AI to humanize, personalize and customize products and services to their clients and expand their outreach and engagement [81]. Hyper personalization is the use of customer data to create and present customized contacts, information, or recommendations to customers. These customizations are created based on individual customer profiles. Profiles rely on data from browsing patterns, purchase histories, geographic location, demographic data, and behavioral data [82].

For example, Thread, a UK-based fashion retailer, offers customers AI-based product recommendations as a “personal stylist”, from information collected from style quizzes and ongoing reactions to product recommendations and they do this with minimal additional effort or staffing [82].

Hilton Hotels currently uses a robot concierge named Connie in its lobbies to greet guests, answer questions and provide concierge-like answers to guests using natural language processing capabilities to interact with guests and develop meaningful profiles [82].

Although Under Armour is known for clothing, they reach customers through lifestyle activities of health and fitness, so they created the Record app, which collects user information on sleep, diet and physical activity. They then create personalized health goals and workout plans and after customers work out will provide feedback on the user’s workout effectiveness to help maximize their future efforts [82].

7.5. Diversity/Inclusivity

While AI is quickly becoming a new tool in the CEO tool belt to drive revenues and profitability, it has also become clear that deploying AI requires careful management to prevent “unintentional but significant damage, not only to brand reputation but, more importantly, to workers, individuals, and society as a whole” [83, p1].

Recent research shows that AI bots and voice assistants promoted unfair gender stereotypes by featuring gendered names, voices, or appearances. In the United States, Siri, Alexa, Cortana, and Google Assistant—which collectively total an estimated 92.4% of U.S. market share for smartphone assistants—have traditionally featured female-sounding voices due to the designers’ innate biases that female voices are more helpful, pleasant and accommodating than male ones [84]. In addition, racial and cultural biases also make it difficult for many people to interact easily with AI assistants around the world [85].

AI chatbots, recruitment software and risk assessment tools in the past caused harm by being racist, gender-biased or selecting the wrong people to put into jail [76]. People may not care how Facebook identifies who to tag in a given picture, but when AI systems are used to make diagnostic suggestions for skin cancer based on automatic picture analysis, understanding how such recommendations have been derived becomes a critical issue [63].

Experts say that AI is still “fragile, opaque, biased and not robust enough” to provide trustworthiness [87]. Leaders need to take the necessary steps to ensure that AI is being used in an ethical manner by consistent reliance on organizational values.

Three ways to accomplish this are: 1) Clarify how values translate into the selection of AI applications, 2) Provide guidance on definitions and metrics used to evaluate AI for bias and fairness, and 3) Prioritize organizational values [83]. Expanding the concept of AI to

'Responsible AI' is essential to ensure fairness, ethics, security/safety, privacy, transparency and accountability issues are considered [88].

"Business leaders may claim that diversity and inclusivity are core goals, but they then need to follow through in the people they hire and the products their companies develop" [19]. Ensuring minorities are well represented among both users and evaluators of AI will make AI more accessible and inclusive [88].

The covid-19 pandemic first discovered in 2019 has accelerated the need for the adoption of digital tools in education, particularly in the science, technology engineering and mathematics (STEM) arena. A majority of software developers are still males with only 25% women in the U.S. and minority racial groups are totally underrepresented in technology fields [89].

The goal is to create a stronger foundation for STEM literacy, inclusion, and diversity of STEM students and preparing the STEM workforce of the future. With the growing demand for advanced skill sets, educators can provide creative and more targeted learning rather than focusing on the repetitive tasks of creating problem sets. The net result is better learning outcomes for a wider group of students and requires collegial partnering, ongoing development, and thorough testing to implement [84].

7.6. Collaboration and Cross Discipline

"Creating differentiated experiences through personalization and immersive education will play a crucial role in the growth of remote learning," said Avasant's Research Leader, Pooja Chopra [53]. "Educational institutions should collaborate with EdTech companies and progressive service providers to accelerate digital transformation" [53].

Bibliometric studies that connect different disciplines are of growing interest in the analysis of the impact of AI synergies and their future within the research community. An example of this is a paper [91] which shows that the structure and model of the scientific production of researchers worldwide and the relationships between quality, references, and synergies among authors increases as collaboration across disciplines is applied. Multi-disciplinary research is vital to effective and natural human-robot connections as well [92].

Interdisciplinary research in artificial intelligence is a way to garner synergistic outcomes across industries from the AI field. To that end, researchers [93] recommend three strategies: 1) Collaborate on ways AI can impact other fields and look to new ideas from other fields to apply to AI; 2) Explain how decisions are made, be transparent about data biases, and use high level evaluators and regulators to evaluate processes; and 3) Scientific and educational experts should increase their AI educational levels.

Human-robot interaction challenges AI in many regards: dynamic, partially unknown environments not originally robot-friendly; a broad variety of situations with rich semantics to understand and interpret; human interactions requiring fine yet socially acceptable control strategies; natural and multi-modal communication requiring common-sense knowledge and divergent mental models. Collaboration of researchers and practitioners from across a variety of fields to integrate and share their data, knowledge, understandings and experiences is essential to meeting these challenges [58]. Cross-functional AI teams made up of diverse participants lead to greater innovations, more collaborations and better outcomes [94].

8. CONCLUSIONS

In this paper outlining a 'Framework for Artificial Intelligence', the authors analyzed the triggers for AI development as well as the influencers to AI adoption. There is no doubt that current triggers such as speed, cost, accuracy, diversity/inclusion, competitiveness, personalization and the need for cross-disciplinary collaboration will continue into the foreseeable future. The present factors such as the coronavirus pandemic of 2019, climate change, customer needs, or resources may fluctuate or change in the future, but there will always be influencers that encourage wider AI adoption and those that discourage AI deployment in organizations. In this comprehensive look at the past, present and future applications in key industry sectors, a better and more comprehensive model for AI emerges.

REFERENCES

- [1] Taherdoost, Hamed (2018) "A review of technology acceptance and adoption models and theories", *Procedia Manufacturing*, Vol. 22, pp960-967.
- [2] Davis, F.D. (1986) *Technology Acceptance Model for empirically testing new end-user information systems: Theory and results*, Ph.D. dissertation, MIT Sloan School of Management, Cambridge: MA.
- [3] Sila, I. (2015) "The state of empirical research on the adoption and diffusion of business-to-business e-commerce", *International Journal of Electronic Business*, Vol. 12, No. 3, pp258-301.
- [4] Venkatesh, V. et al., (2003) "User acceptance of information technology: Towards a unified view", *MIS Quarterly*, Vol. 27, No. 3, pp425-478.
- [5] Paul, K.J., Musa, M. & Nansubuga, A. K. (2015) "Facilitating condition for e-learning adoption – Case of Ugandan universities", *Journal of Communication and Computer*, Vol. 12, pp244-249
- [6] Carter, L. & Bélanger, F. (2005) "The utilization of e-government services: Citizen trust, innovation and acceptance factors", *Information Systems Journal*, Vol. 15, No. 1, pp5-26.
- [7] Zarkout, Bassam (2019) "The DX journey in the enterprise and its leadership", *IIC Journal of Innovation*, Vol. 18, pp19-34. <https://viewer.joomag.com/18th-edition-of-the-journal-of-innovation-rapid-advancements-in-digital-transformation/0056845001637120709?page=26>
- [8] Allam, Z., Dey, G. & Jones, D.S. (2020) "Artificial Intelligence (AI) provided early detection of the coronavirus (COVID-19) in China and will influence future urban health policy internationally", *Artificial Intelligence*, Vol. 1, pp156-165.
- [9] Osikoya, R. (2020) "How we can use tech to improve diversity in the workplace", *World Economic Forum*, June 23, <https://www.weforum.org/agenda/2020/06/technology-ally-inclusion-diversitywork/#:~:text=Technology%20can%20be%20an%20enabler%20of%20greater%20diversity,our%20thinking%2C%20influence%20processes%20and%20ultimately%20change%20behaviours.>
- [10] Sadiq, R. B., Safie, N., Rahman, A. H. & Goudarzi, S. (2021) "Artificial intelligence maturity model: A systematic literature review", *Peer Journal of Computer Science*, Vol. 7, ppe661. <https://doi.org/10.7717/peerj-cs.661>
- [11] Asfaw, S., Battista, F. D., & Lipper, L. (2016) "Agricultural technology adoption under climate change in the Sahel: Micro-evidence from Niger", *Journal of African Economies*, Vol. 25, No. 5, pp637-669. <https://doi.org/10.1093/jae/ejw005>
- [12] Sahu P. (2020, April 4) "Closure of universities due to coronavirus disease 2019 (COVID-19): Impact on education and mental health of students and academic staff", *Cureus*, Vol. 12, No. 4, e7541. <https://doi.org/10.7759/cureus.7541>
- [13] Pokhrel, S., & Chhetri, R. (2021) "A literature review on impact of COVID-19 pandemic on teaching and learning", *Higher Education for the Future*, Vol. 8, No. 1, pp133-141, <https://doi.org/10.1177/2347631120983481>
- [14] Saxena, Kritika. (2018, January 5) "Five on-point reasons why businesses are adopting Artificial Intelligence", *Resourcifi*, Accessed on December 11, 2021 at <https://www.resourcifi.com/blog/5-point-reasons-businesses-adopting-artificial-intelligence-outgrow/>
- [15] Jöhnk, J., Weißert, M. & Wyrski, K. (2021) "Ready or not, AI comes – An interview study of organizational AI readiness factors", *Business Information System Engineering*, Vol. 63, pp5-20. <https://doi.org/10.1007/s12599-020-00676-7>.

- [16] Turing, Alan (1950) "Computing machinery and intelligence", *Mind*, Vol. 59, No. 236, pp433-460.
- [17] Russell, S.J., & Norvig, P. (2010) *Artificial Intelligence: A Modern Approach*, (3rd ed.). Upper Saddle River, New Jersey, Prentice-Hall.
- [18] Campbell, M., Hoane, A. J. & Hsu, Feng-Hsiung (2002) "Deep Blue," *Artificial Intelligence*, Vol. 134, No. 1-2, pp57-83.
- [19] Popenici, S. A. D. & Kerr, S. (2017) "Exploring the impact of Artificial Intelligence on teaching and learning in higher education", *Research and Practice in Technology Enhanced Learning*, Vol. 12, No. 1, pp1-13. doi: <http://dx.doi.org/10.1186/s41039-017-0062-8>
- [20] Ruiz-Real, J., Uribe-Toril, J., Torres Arriaza, J. A. & Jaime de, P. V. (2020) "A look at the past, present and future research trends of Artificial Intelligence in agriculture", *Agronomy*, Vol. 10, No. 11, p1839. doi: <http://dx.doi.org/10.3390/agronomy10111839>
- [21] King, B.A., Hammond, T. & Harrington, J. (2017) "Disruptive technology: Economic consequences of Artificial Intelligence and the robotics revolution", *Journal of Strategic Innovation and Sustainability*, Vol. 12, No. 2, pp53-67.
- [22] Haenlein, M. & Kaplan, A. (2019) "A brief history of Artificial Intelligence: On the past, present, and future of Artificial Intelligence", *California Management Review*, Vol. 61, No. 4, pp5-14. doi:10.1177/0008125619864925
- [23] Bharwani, A. (2017, Sep 18) "10 ways artificial intelligence will impact education sector", *Business World*, Accessed November 22, 2021. <http://bwpeople.businessworld.in/article/10-ways-Artificial-Intelligence-will-impact-education-sector/18-09-2017-126428/>
- [24] Nicola, C. B. & Dalessio, D. (2019) "Artificial Intelligence and the impact on business curricula", *Academy of Business Research Journal*, Vol. 3, pp30-53.
- [25] Araya, D. (2019, January 02). "Who will lead in the age of Artificial Intelligence?", *Forbes*, Accessed May 26, 2019. <https://www.forbes.com/sites/danielaraya/2019/01/01/who-will-lead-in-the-age-of-artificial-intelligence/#3062b7856f95>
- [26] McKinion, J.M. & Lemmon, H.E. (1985) "Expert systems for agriculture", *Computers and Electronics in Agriculture*, Vol. 1, No. 1, pp31-40.
- [27] Attonaty, J.M., Chatelin, M.H. & Garcia, F. (1999) "Interactive simulation modeling in farm decision-making", *Computers and Electronics in Agriculture*, Vol. 22, pp157-170.
- [28] Kohzadi, N., Boyd, M., Kermanshahi, B. & Kaastra, I. A. (1996) "Comparison of artificial neural network and time series model for forecasting commodity prices," *Neurocomputing*, Vol. 10, No. 2, pp169-181.
- [29] Nair, B. & Mohandas, V. (2015) "Artificial Intelligence applications in financial forecasting: A survey and some empirical results", *Intelligent Decision Technologies*, Vol. 9, pp99-140.
- [30] Bryceson, K. & Slaughter, G. (2009) "Integrated autonomy: A modeling-based investigation of agrifood supply chain performance. In *Proceedings of the 2009 11th International Conference on Computer Modelling and Simulation*, Cambridge, UK, March 25-27.
- [31] Hewitson, B.C. & Crane, R.G. (2002) "Self-Organizing Maps: Applications to synoptic climatology", *Climate Research*, Vol. 22, pp13-26.
- [32] Mellit, A. (2008) "Artificial Intelligence technique for modelling and forecasting of solar radiation data: A review", *International Journal of Artificial Intelligence Soft Computing*, Vol. 1, pp52-76.
- [33] Murase, H. (2000) "Artificial Intelligence in agriculture", *Computers and Electronics in Agriculture*, Vol. 29, pp1-20.
- [34] Kaur, M., Gulati, H. & Kundra, H. (2014) "Data mining in agriculture on crop price prediction: Techniques and applications", *International Journal of Computational Applications*, Vol. 99, No. 12, pp1-2.
- [35] Li, G. Q.; Xu, S. W.; & Li, Z. M. (2010) "Short-term price forecasting for agro-products using Artificial Neural Networks", *Agricultural Seletivo Processes*, Vol. 1, pp278-287.
- [36] Dahikar, S. S. & Rode.V. S. (2014) "Agricultural crop yield prediction using artificial neural network approach", *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, Vol. 2, No. 1, pp683-686.
- [37] Aznar-Sánchez, J. A., Piquer Rodríguez, M., Velasco-Muñoz, J. F. & Manzano-Agugliaro, F. (2019) "Worldwide research trends on sustainable land use in agriculture." *Land Use Policy*. Accessed May 10, 2020. doi: 10.1016/j.ecolind.2018.12.045
- [38] El Yasmine, A. S. L., Ghani, B. A., Trentesaux, D. & Bouziane, B. (2014) "Supply chain management using multi-agent systems in the agri-food industry", *Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics*, pp145-155.

- [39] Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Hellmann Santos, C. & Pekkeriet, E. (2020) "Agricultural robotics for field operations", *Sensors*, Vol. 20, p2672.
- [40] Lowenberg-DeBoer, J., Huang, I. Y., Grigoriadis, V. *et al.* (2020) "Economics of robots and automation in field crop production", *Precision Agriculture*, Vol. 21, pp278-299.
- [41] Ren G, Lin T, Ying Y, Chowdhary G, Ting, K. C. (2020) "Agricultural robotics research applicable to poultry production: A review", *Computer and Electronics in Agriculture*, Vol. 169, e105216. <https://doi.org/10.1016/j.compag.2020.105216>
- [42] Appeltans, S., Pieters, J. G. & Mouazen, A. M. (2021) "Potential of laboratory hyperspectral data for in-field detection of *Phytophthora infestans* on potato", *Precision Agriculture*, <https://doi.org/10.1007/s11119-021-09865-0>
- [43] Maloku D., Balogh, P., Bai, A., Gabnai, Z. & Lengyel, P. (2020) "Trends in scientific research on precision farming in agriculture using science mapping method", *International Review of Applied Sciences and Engineering*, Vol. 11, pp232–242.
- [44] Rose, D. C., Wheeler, R., Winter, M., Lobley, M., & Chivers, C.A. (2020) "Agriculture 4.0: Making it work for people, production, and the planet", *Land Use Policy*, Vol. 100, e104933.
- [45] Roy, S. K. & De, D. (2020) "Genetic algorithm-based Internet of Precision Agricultural Things (IopaT) for agriculture 4.0", *Internet of Things 2020*, e100201. <https://doi.org/10.1016/j.iot.2020.100201>
- [46] Kuhl, L. (2019) "Technology transfer and adoption for smallholder climate change adaptation: opportunities and challenges", *Climate and Development*, Vol. 12, No. 4, pp353-368.
- [47] Ryan, M. (2020) "Agricultural big data analytics and the ethics of power", *Journal of Agricultural and Environmental Ethics*, Vol. 33, pp49-69.
- [48] Mokarram, M. & Khosravi, M.R. (2021) "A cloud computing framework for analysis of agricultural big data based on Dempster Shafer theory", *Journal of Supercomputing*, Vol. 77, pp2545–2565.
- [49] Zhai, F. Z., FernánMartínez, J., Beltran, V., & Martínez, N. L. (2020) "Decision support systems for agriculture 4.0: Survey and challenges", *Computers and Electronics in Agriculture*, Vol. 170, No. 77, pp2545-2565.
- [50] Dorn, D. S. (1989) "Simulation games: One more tool on the pedagogical shelf", *Teaching Sociology*, Vol. 17, No. 1, pp1–18. <https://doi.org/10.2307/1317920>.
- [51] Psotka, J. & Mutter, S. A. (1988) *Intelligent Tutoring Systems: Lessons Learned*, Mahwah, New Jersey, Lawrence Erlbaum Associates.
- [52] Rollinger, Christian (2020, January 9) *Classical Antiquity in Video Games: Playing with the Ancient World*, New York, New York, Bloomsbury Publishing.
- [53] RadarView™ report. (2021, Aug 19). PR Newswire <https://www.proquest.com/wire-feeds/remote-learning-strategic-priority-educational/docview/2562526207/se-2?accountid=10139>
- [54] Rivers, K. & Koedinger, K. R. (2017) "Data-driven hint generation in vast solution spaces: A self-improving Python programming tutor", *International Journal of Artificial Intelligence Education*, Vol. 27, pp37-64.
- [55] "5 EdTech trends shaping business education from Artificial Intelligence to virtual reality", (2016, June 17) *ICT Monitor Worldwide*, e1797581 <https://www.proquest.com/wire-feeds/5-edtech-trends-shaping-business-education/docview/1797581583/se-2?accountid=10139>
- [56] Gibney, E. (2017) "Google secretly tested AI bot", *Nature*, Vol. 541, No. 7636, p142.
- [57] Garrido, A. (2012) "AI and mathematical education", *Education Sciences*, Vol. 2, No.1, pp22-32. <https://doi.org/10.3390/educ2010022>
- [58] Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., & Alami, R. (2017) "Artificial cognition for social human–robot interaction: An implementation", *Artificial Intelligence*, Vol. 247, pp45–69.
- [59] Benke, K. & Benke, G. (2018) "Artificial Intelligence and big data in public health", *International Journal of Environmental Research and Public Health*, Vol. 15, No. 12, p2796.
- [60] Middleton, B., Sittig, D. F., & Wright, A. (2016) "Clinical decision support: A 25-year retrospective and a 25-year vision", *Yearbook Medical Informatics*, Vol. 1, pp5103-5116.
- [61] Khanna, S., Sattar, A., & Hansen, D. (2013) "Artificial Intelligence in health—The three big challenges", *Australasian Medical Journal*, Vol. 6, No. 5, pp315-317.
- [62] López-Robles, J. R., Otegi-Olaso, J. R., Gómez, I. P., & Cobo, M. J. (2019) "Thirty years of intelligence models in management and business: A bibliometric review", *International Journal of Information Management*, Vol. 48, pp22–38.
- [63] Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A. Kalloo, A. Ben Hadj Hassen, A., Thomas, L., Enk, A. & Uhlmann, L. (2018) "Man against machine: Diagnostic

- performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists”, *Annals of Oncology*, Vol. 29, No. 8, pp1836-1842.
- [64] Cascianelli S., Scialpi M., Amici, S., Forini, N., Ministrini, M., Fravolini, M. L., Sinzinger, H., Schillaci, O. & Palumbo, B. (2017) “Role of Artificial Intelligence techniques (Automatic Classifiers) in molecular imaging modalities in neuro- degenerative disease”, *Current Alzheimer Research*, Vol. 14, No. 2, pp198-207.
- [65] Steinhubl, S.R., & Topol, E.J. (2015) “Moving from digitalization to digitization in cardiovascular care: why is it important, and what could it mean for patients and providers?”, *Journal of American College of Cardiology*, Vol. 66, pp1489-1496.
- [66] Kubota, K.J., Chen, J.A., & Little, M.A. (2016) “Machine learning for large-scale wearable sensor data in Parkinson’s Disease: Concepts, promises, pitfalls, and features”, *Movement Disorders*, Vol. 31, No. 9, pp1314-1326.
- [67] Levy, S., Duda, M., Haber, N., & Wall, D. P. (2017) “Sparsifying machine learning models identify stable subsets of predictive features for behavioral detection of autism”, *Molecular Autism*, Vol. 8, pp65-82.
- [68] Hewitson, L. Mathews, J. A., Devlin, M., Schutte, C., Lee, J. & Germain, D. C. (2021) “Blood biomarker discovery for Autism Spectrum Disorder: A proteomic analysis”, *PLoS ONE*, Vol. 16, No. 2, pp1-15.
- [69] Jalali, A., Simpaio, A.F., Gálvez, J.A., Licht, D. J., & Nataraj, C. (2018) "Prediction of periventricular leukomalacia in neonates after cardiac surgery using machine learning algorithms”, *Journal of Medical Systems*, Vol. 42, pp177-193.
- [70] Shu, LQ., Sun, YK., Tan, L. H., Shu, Q. & Chang, A. C. (2019) “Application of Artificial Intelligence in pediatrics: Past, present and future”, *World Journal of Pediatrics*, Vol. 15, pp105-108.
- [71] Jiang F, Jiang Y, Zhi H, et al. (2017). Artificial Intelligence in healthcare: Past, present and future”, *Stroke and Vascular Neurology*, Vol. 2. e000101. doi:10.1136/svn-2017-000101
- [72] Hamet, P., & Tremblay, J. (2017) “Artificial Intelligence in medicine”, *Metabolism*, Vol. 69, ppS36-S40.
- [73] Aggarwal, R. (2020, February 26) “Top 10 AI Trends to Watch Out For in 2020”, Accessed October 11, 2021. <https://datafloq.com/read/top-10-ai-trends-to-watch-out-2020/7813>
- [74] SnapChat (2021, February), “Employees want more AI in the workplace”, *SnapLogic*, Accessed December 4, 2021. <https://www.snaplogic.com/resources/infographics/employees-want-more-ai-in-the-workplace>
- [75] Wiggers, K. (2021, July) “Employees want more AI to boost productivity, study finds”, *The Machine: Making Sense of AI*, Assessed on October 10, 2021 at <https://venturebeat.com/2021/07/20/employees-want-more-ai-to-boost-productivity-study-finds/>
- [76] Carass, A., Cuzzocreo, J. L., Han, S., Hernandez-Castillo, C. R., Rasser, P. E., Ganz, M., et al. (2018) “Comparing fully automated state-of-the-art cerebellum parcellation from magnetic resonance images”, *Neuroimage*, Vol. 183, pp150-172.
- [77] Su H., Shen, Y., Xing F, Qi, X., Hirshfield K. M., Yang, L., & Foran, D. J. (2015) “Robust automatic breast cancer staging using a combination of functional genomics and imageomics”, *Conference Proceedings IEEE Engineering in Medicine and Biology Society*, pp7226-7229.
- [78] Kaplan, Susan (2021, December 3). Interview on collaboration between Modal Technology and McGill University Health Center Research.
- [79] Akula, Vasudeva (2021) Three ways AI can protect and bring costs down during challenging times. *Forbes*, Accessed December 4, 2021. <https://www.forbes.com/sites/forbestechcouncil/2020/07/20/three-ways-ai-can-protect-revenue-and-bring-costs-down-during-challenging-times/?sh=534447a33153>
- [80] Spatharou, A. Hieronimus, S. & Jenkins, J. (2020, March 10) “Transforming healthcare with AI: The impact on the workforce and organizations”, *McKinsey & Company Executive Briefing*, Accessed on August 5, 2020 at <https://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/transforming-healthcare-with-ai>
- [81] Salmon-Powell, Z., Scarlata, J., & Vengrouskie, E. F. (2021) “Top five Artificial Intelligence trends affecting leadership & management”, *Journal of Strategic Innovation and Sustainability*, Vol. 16, No. 4, pp1-3.
- [82] Maayan, Gilad (2021) “Hyper personalization: Customizing service with AI”, *IEEE Computer Society*, Accessed November 4, 2021, <https://www.computer.org/publications/tech-news/trends/hyper-personalization-customizing-service-with-ai/>

- [83] Burkhardt, R., Hohn, N., & Wigley, C. (n.d.) "Leading your organization to responsible AI", *McKinsey and Company*. Accessed May 25, 2019, https://www.washingtonpost.com/sports/2019/07/10/baseballs-robot-umpires-are-here-you-might-not-even-notice-difference/?utm_term=.5a947730d227
- [84] Chin, C. & Robison, M. (2020, November 23) "How AI bots and voice assistants reinforce gender bias", *Brookings*, Accessed December 9, 2021, <https://www.brookings.edu/research/how-ai-bots-and-voice-assistants-reinforce-gender-bias/>
- [85] Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M. Mengesha, Z., Toups, C., Rickford, J. R., Jurafsky, D. & Goel, S. (2020) "Racial disparities in automated speech recognition", *Proceedings of the National Academy of Sciences*, Vol. 117, No. 14, pp7684-7689. <https://doi.org/10.1073/pnas.1915768117>
- [86] Dressel J, & Farid H. (2018, January 17) "The accuracy, fairness, and limits of predicting recidivism", *Scientific Advances*, Vol. 4, No. 1, eaao5580. <https://doi.org/10.1126/sciadv.aao5580>.
- [87] Vergun, D. (2021) "Artificial Intelligence is a work in progress, official says", *Defense OneGenius Machines 2021 Summit*, Accessed September 9, 2021. <https://www.defense.gov/News/News-Stories/Article/Article/2480288/artificial-intelligence-is-a-work-in-progress-official-says>
- [88] Arrieta, A., Diaz-Rodriguez, N., Del Ser, J. Benítez, A., Tabik, S., Barbado, A. et al. (2020) "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI", *Information Fusion*, Vol. 58, pp82-115.
- [89] Daugherty, P. R., Wilson, H. J., & Chowdhury, R. (2019) "Using Artificial Intelligence to promote diversity", *MIT Sloan Management Review*, Vol. 60, No. 2, p1-19. [80]
- [90] Krasovskiy, Dmitry (2020, November 24) "The challenges and benefits of adopting AI in STEM Education", *Upjourney*, Accessed December 10, 2021. <https://upjourney.com/the-challenges-and-benefits-of-adopting-ai-in-stem-education>
- [91] Gu, Y. (2004) "Global knowledge management research: A bibliometric analysis", *Scientometrics*, Vol. 61, pp171-190.
- [92] Kragic, Danica & Sandamirskaya, Yulia (2021) "Effective and natural human-robot interaction requires multidisciplinary research", *Science Robotics*, Vol. 6, No. 58, eabc 7022.
- [93] Kusters, R., Misevic, D. Berry, H. Cully, A., Le Cunff, Y., Dandoy, L., et al. (2020, November 3) "Interdisciplinary research in Artificial Intelligence: Challenges and opportunities", *Frontiers in Big Data*, article 577974. <https://doi.org/10.3389/fdata.2020.577974>
- [94] Quilici, E. (2021) "How cross functional interactions can boost collaboration", *Pharmaceutical Executive*, Vol. 41, No. 3, pp1-8.

AUTHORS

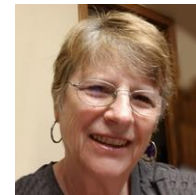
Professor Richard A. Fulton (M.S., Illinois State University) has taught full time computer science and information systems courses at Troy University – e campus for the past 18 years and previously at Illinois State University. His articles have been published in *The Journal of Technology Research*, *The Journal of Scientific Information on Political Theory*, *Developments in Business Simulations and Experiential Learning*, and the *International Journal of Innovation, Technology and Management*.



Dr. Diane J. Fulton (Ph. D., University of Tennessee-Knoxville) is Emeritus Professor of Management at Clayton State University, located in Morrow, Georgia. Her research interests include advanced technologies, innovations and online teaching tools. She has published several books, book chapters, and numerous articles in academic journals, including *California Management Review*, *Planning Review*, *Journal of Small Business Management*, *International Journal of Management Education and Entrepreneurship Theory and Practice*.



Susan Kaplan (BAS, MAS, University of Minnesota-Duluth) is Executive Vice President and Chief Management Officer of Modal Technology Corporation, a high-tech firm located in Minneapolis, Minnesota that offers new and proven solutions for artificial intelligence and machine learning. Ms. Kaplan is a Founder and Director at RISC AI and was Founder and President for Quality Management Systems. She aided organizations in healthcare, government, manufacturing, and service sectors to improve profitability. She is the author of *The Grant Writing Process*.



AN EVALUATION DATASET FOR LEGAL WORD EMBEDDING: A CASE STUDY ON CHINESE CODEX

Chun-Hsien Lin and Pu-Jen Cheng

Department of Computer Science & Information Engineering,
National Taiwan University, Taipei, Taiwan

ABSTRACT

Word embedding is a modern distributed word representations approach and widely used in many natural language processing tasks. Converting the vocabulary in a legal document into a word embedding model facilitates subjecting legal documents to machine learning, deep learning, and other algorithms and subsequently performing the downstream tasks of natural language processing vis-à-vis, for instance, document classification, contract review, and machine translation. The most common and practical approach of accuracy evaluation with the word embedding model uses a benchmark set with linguistic rules or the relationship between words to perform analogy reasoning via algebraic calculation. This paper proposes establishing an 1,134 Legal Analogical Reasoning Questions Set (LARQS) from the 2,388 Chinese Codex corpus using five kinds of legal relations, which are then used to evaluate the accuracy of the Chinese word embedding model. Moreover, we discovered that legal relations might be ubiquitous in the word embedding model.

KEYWORDS

Legal Word Embedding, Chinese Word Embedding, Word Embedding Benchmark, Legal Term Categories.

1. INTRODUCTION

Word embedding is a modern distributed word representations approach and widely used in many natural language processing tasks, such as semantic analysis [1], text classification [2], and machine translation [3, 4]. Most general-purpose applications mentioned above apply the evaluation benchmark dataset released by Google to assess their word embedding. For Chinese, the evaluation benchmark dataset was translated from the former by the Chinese Knowledge and Information Processing group (CKIP group) of the Academia Sinica Institute of Information Science. However, the general-purpose word embedding model cannot fully meet the task requirements of specific fields, such as biomedicine, financial opinion mining, the legal profession, etc. All have their own idiosyncratic terminology and thus should not be included in general-purpose word embedding models. In fact, in developing applications in these specific fields, collecting domain-relevant textual data to build a dataset and establish a word embedding model is often necessary. On the other hand, various languages also have their linguistic particularities. Sometimes, at the lexical word level, there will be no satisfactory translation. For example, plural nouns and verb tenses in English have no lexical equivalent in Chinese. Therefore, translating the existing general-purpose evaluation benchmark dataset directly as the target language word embedding evaluation benchmark has its limitations. As such, in addition to building a word embedding model for a specific field, it is very challenging to develop a

corresponding evaluation benchmark dataset to evaluate the pros and cons of the word embedding model trained on a corpus of a specific field.

As the traditional Chinese text dataset lacked legal documents, we collected numerous articles containing legal provisions to serve as our dataset. Next, we established the basis of the legal relations dataset of this text dataset by conducting field expert review and induction. Finally, based on this dataset, we established an evaluation benchmark dataset for analogy reasoning. Using manual methods to create datasets is a last resort because of the serious drawbacks of higher costs, time expenditure, observer bias, and the potentially small size of the dataset, along with its innate irreproducibility. (We leave it to the future to develop an automatic method.) While many datasets are available for Chinese NLP experimentation, it is not easy to find legal domain experts to review and establish them. Given the absence of a Chinese legal data collection and evaluation benchmark dataset in our domestic environment, the manual method is our only recourse. While time and energy-consuming, the latter is nonetheless a pretty direct and intuitive approach. As regards the Chinese legal word embedding model and the evaluation benchmark dataset with legal relations, it will be very helpful in the future to introduce state-of-art technologies such as machine learning and deep learning into the Chinese legal profession application.

All the base data used in this experiment is collected from the 2,388 code articles found in the “Laws and Regulations Database of The Republic of China” (全國法規資料庫). Prior to the field expert manually reviewing the legal relationship between words in our vocabulary list, we use CKIPTagger, released by the CKIP group (aka CKIP Lab), to segment the sentences in the dataset as well as TF-IDF to sort and view the vocabulary. We also deploy gensim to combine the codex datasets mentioned above. Next, two modes, skip-gram and CBoW, respectively, are used to generate word embedding models. Then, TensorBoard is used to visually assist manual inspection of the relevance of each word in the vector space. Finally, we manually select the appropriate legal relationship vocabulary to build the legal analogy reasoning questions (Legal Analogical Reasoning Questions Set, LARQS) in order to evaluate the accuracy of the word embedding model trained from the codex corpus.

Various evaluation methods have been proposed to assess the qualities of word embedding models [5]. However, there is still no scientific consensus on which evaluation method should be used for word embedding models [6]. Therefore, this paper applied the simple algebraic calculation method as analogical reasoning proposed by Mikolov [7] to evaluate the accuracy of word embedding models. Applying Mikolov’s simple algebraic calculation method is intuitive, but also requires an evaluation benchmark dataset with broader relationship coverage for the accuracy of the evaluation to be more objectively determined. As such, this paper attempts to establish an 1,134 Legal Analogical Reasoning Questions Set (LARQS) with five categories of legal relationships from the corpus collecting 2,388 Chinese legal codices so as to evaluate the accuracy of Chinese word embedding from the perspective of legal relations.

Our main contributions in this work can be summarized as follows:

- We collected the legal provisions of 2,388 laws and regulations promulgated and implemented in Taiwan as a codex corpus for experimentation. Next, we trained a legal word embedding model based on this corpus and released it to the public for future research.
- We established and released a 1,134 Legal Analogical Reasoning Questions Set (LARQS) with five categories of legal relationships from the corpus mentioned above to evaluate the accuracy of Chinese word embedding from the perspective of legal relations.

- We assessed the accuracy of several word embedding models with the Google evaluation benchmark dataset. Our conclusion is that, generally speaking, the evaluation benchmark dataset is not suitable for the legal profession.
- We discovered that legal relations may be ubiquitous in the word embedding model.

This paper is organized as follows: Section 2 provides an essential background for Chinese word segmentation, which is the first and most critical step in Chinese natural language processing, and reviews the development and research of word embedding models in Chinese. Moreover, we review several benchmark datasets for the word embedding model and the evaluation approach used in this paper. In section 3, we give a brief description of the dataset source of the experiment, the established procedures, and the tools used in this paper. Section 4 explains the meanings implied by legal terms and their interrelationship on the conceptual level. Section 5 describes the evaluation dataset proposed in this paper and the experimental results of other evaluation datasets on different Chinese word embedding models. Finally, section 6 discusses the possibility of working with the LARQAS benchmark dataset in future.

2. RELATED WORKS

The most famous approach to the distributed representation of words in a vector space is the word embedding model proposed by Mikolov [8]. Capable of quickly producing compact vectors, this algorithm is widely used by the natural language processing community. On the other hand, when it comes to the distributed representation of words, simple algebraic calculations can be used to obtain the offset value of each word pair. This approach can achieve the effect of acquiring “linguistic regularities” by analogical reasoning, as in “king - man + woman \approx queen”. Given the simplicity of this calculation, some scholars have also tried to capture the semantics of biomedical concepts [9], and extract mentions of adverse drug reactions from informal text found on social media [10]. In addition to obtaining semantic rules, we can also use the vector calculation method described above to induce morphological transformations between words [11]. Moreover, Ash [44] has similarly shown that the word embedding model trained on a corpus of statutes result in $\text{vector}(\text{"corporate income tax"}) - \text{vector}(\text{"corporation"}) + \text{vector}(\text{"person"}) \approx \text{vector}(\text{"personal income tax"})$. Based on the premise that these words’ “linguistic regularities” are assumed to have linear relations, vector offset is used to capture the syntactic and semantic rules of word analogy reasoning. However, the semantic coverage of this approach is limited [12, 13]. This is hardly surprising because even Google’s public evaluation benchmark dataset contains only 14 linguistic categories.

Although there has been considerable research into the evaluation of Chinese word embedding, most of it has been based on the aforementioned algebraic calculation as analogy reasoning of the word embedding vector offset value [14], which facilitates exploring the morphology or semantics of Chinese words. Research into the concepts and correlation between legal terms and word embedding models has been comparatively rare. Factors contributing to this include the difficulty of obtaining datasets and appropriate field experts to participate in such research.

When it comes to Chinese natural language processing workflow, the first obstacle is Chinese word segmentation [15]. Moreover, the accuracy of Chinese word segmentation results has varying degrees of impact on subsequent downstream Chinese natural language processing tasks [16]. To reduce the error rate in these tasks, it is thus imperative to choose the most appropriate Chinese word segmentation tools. In this regard, many researchers have already produced numerous outstanding study findings in this field [17]. As such, for this paper we were able to select the most appropriate available tools to handle Chinese word segmentation.

To assess and compare the performance of the benchmark set, we build two sets of word embedding models, one with codex articles, the other with the Chinese version of Wikipedia. After establishing the word embedding model, the next step is to explore means for evaluating its accuracy. Subsequently, we assess the accuracy of candidate word embedding models using the Google evaluation benchmark dataset translated by the CKIP Group of the Academia Sinica Institute of Information Science, the CA8 evaluation benchmark dataset released by Shen Li [14], and the legal analogy dataset proposed in this paper (Legal Analogical Reasoning Questions Set, LARQS). Using these evaluation benchmark datasets, we then apply the simple algebraic calculation analogical reasoning approach proposed by Mikolov [7] and compare it with these benchmarks and word embedding models.

2.1. Chinese Word Segmentation

Chinese word segmentation is the first and most critical step in Chinese natural language processing. Many researchers have proposed methods to solve this necessary process [18-20]. Moreover, a deep neural network has recently been introduced to work on this issue and achieved conspicuously superior results [21]. There are many Chinese word segmentation tools. Python is the primary programming language for this paper's experiment. In the early stage of the experiment, we used the word segmentation tool "Jieba", a famous Chinese word segment package for Python whose primary advantage is speed. However, Jieba's development was based on simplified Chinese. When processing large numbers of traditional Chinese documents, it turned out to be outperformed by CKIPTagger (released by the Chinese Knowledge Information Processing Group of the Academia Sinica Institute of Information Science). With our objective being to handle Chinese codex articles, and the words in the codex belonging to the legal professional field, Jieba's word segmentation results proved even more unsatisfactory. After manually reviewing the segment results of the word segmentation tools Jieba and CKIPTagger, we finally chose CKIPTagger as the word segmentation tool for this experiment because of its superior accuracy with traditional Chinese word segments. Figure 1 is an example of word segmentation in a Chinese law article.

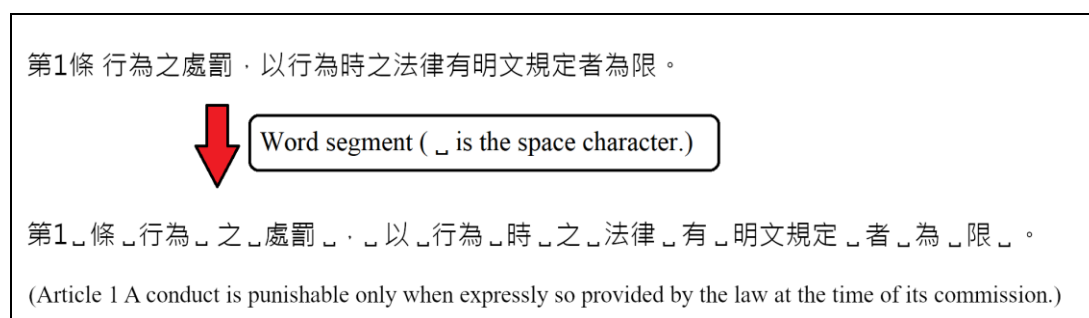


Figure 1. Word segment a Chinese law article with a space character.

2.2. The Model Architectures for Word Embedding

In many Natural Language Processing tasks, the representation of a lexical word into a numerical form that the computer can calculate is a fundamental and necessary pre-processing procedure. The conventional one-hot vector is an intuitive approach, but the distributed word representations can represent lexical words with a low-dimensional dense vector and embed more latent information [22]. Distributional word representations based on co-occurrence information between lexical words are LSA [23] and LDA [24]. There exist various approaches for obtaining the distributed dense real-valued vector from unlabelled text corpora. The most famous of which

are Glove [25] and word2vec Skip-gram [7]. Recent work has shown that transforming lexical words into a distributed dense real-valued vector in a geometric space can capture the semantic "meaning" of a single word embedded in the vector via simple algebraic calculations and other methods. [26-29].

The distributed word embedding architectures proposed by Mikolov are Skip-gram and CBoW. Due to its success in modelling English documents, word embedding has been applied to Chinese text. Benefiting from the internal structural information of Chinese characters, many studies tried to enhance the quality of Chinese word embeddings with radicals [30-32], sub-word components [33, 34], glyph features [35], strokes [36], and pronunciation [37]. To limit the scope of this paper, we choose Skip-gram because, after comparing the word embedding model established by the two corpora used in this experiment, we found Skip-gram to have the best performance on average.

2.3. Existing public benchmark datasets and finding similar words

As regards evaluating word embedding models, Google has released a test set consisting of about 20,000 questions-words plus their syntactic and semantic relations. However, it only offers morphological and semantic relations for both analogical reasoning and capturing linguistic regularities, but does not explore other relationships in great breadth. For example, as mentioned above, the semantic relations of biomedical concepts or legal term relations were not included in the Google test set. The test set has a total of 14 relationships, encompassing 9 morphological and 5 semantic categories, and a total of 19,544 questions. The Chinese Knowledge Information Processing Group of the Academia Sinica Institute of Information Science (aka CKIP Group) also translated the Google test set [38], with 11,126 questions. A summary of the content released by Google and translated by the CKIP Group is shown in Table 1:

Table 1. A summary of the benchmark released by Google and translated by the CKIP Group

Relation	Example	Translated to Chinese
capital-common-countries	Beijing China Tokyo Japan	北京 中國 東京 日本
capital-world	Ankara Turkey Cairo Egypt	安卡拉 土耳其 開羅 埃及
Currency	USA Dollar Europe Euro	美國 美元 歐洲 歐元
city-in-state	Chicago Illinois Honolulu Hawaii	芝加哥 伊利諾州 檀香山 夏威夷州
family	brother sister king queen	兄弟 姐妹 國王 皇后
gram1-adjective-to-adverb	amazing amazingly	(No Chinese mapping)
gram2-opposite	decided undecided efficient inefficient	決定 未決 有效率 低效率
gram3-comparative	bad worse	(No Chinese mapping)
gram4-superlative	bad worst	(No Chinese mapping)
gram5-present-participle	code coding	(No Chinese mapping)
gram6-nationality-adjective	Albania Albanian China Chinese	阿爾巴尼亞 阿爾巴尼亞人 中國 中國人
gram7-past-tense	dancing danced	(No Chinese mapping)

gram8-plural	banana bananas	(No Chinese mapping)
gram9-plural-verbs	decrease decreases	(No Chinese mapping)

Leveraging word analogical reasoning as an evaluation benchmark is fascinating and has potential as an approach for discovering linguistic relations [39]. In 2010, Turney and Pantel proposed an extensive survey of tasks that could be considered as a reference for measuring the performance of word embedding [40]. In 2013, Mikolov showed that proportional analogies (a is to b as c is to d) could be solved by finding the vector closest to the hypothetical vector calculated as $c - a + b$ (i.e., king - man + woman \approx queen). The assumption is that a “well-trained” word embedding encodes linguistic relations so that they are identifiable via linear vector offset [7, 41]. Although various evaluation methods are mentioned in the literature above, however, to limit the scope of this paper, we choose the original approach proposed by Mikolov to find out the hidden vector d. This simple calculation is quite helpful for subsequent manually finding similar words to build the LARQS benchmark dataset.

3. DATASET CONSTRUCTION AND TOOLS USED IN THE EXPERIMENT

The dataset (Chinese codex dataset) established in this experiment is collected from the Chinese provisions of domestic laws and regulations published in the “Laws and Regulations Database of The Republic of China” (全國法規資料庫). We collected 2,388 statutes and regulations, consisting of 73,365 articles (excluding the article section names, the total is 67,727 articles). Next, word segmentation with CKIPTagger and terms with a word frequency of less than five are manually checked. Ultimately, a dataset emerges consisting of 5,830,894 words. Additionally, using the dataset included in the Chinese version of Wikipedia up to December 1, 2019 and the same parameters as in the following, we established 19 different dimensional word embedding models for comparison.

The tool used to build the Chinese word embedding models in this paper is gensim. Even using this existing tool, the hyper parameters for the training word embedding model greatly influence the final quality [42, 43]. Therefore, to explore the best possible hyper parameters, we selected several hyper parameters such as the vector size of word embedding, iterations, and the window sizes for the experiment. We set the following parameters to train it: vector size from 100 to 1000 dimensions, with an interval of every 50 dimensions, and the number of iterations being 1, 10, 100, 200, and 1000. Chiefly using the skip-gram architecture, and with the word window size being 7 and the smallest token size set as a single Chinese character, we built 19 Chinese word embedding models using these parameters from the dataset mentioned above.

Figure 2 is a schematic illustration of the construction process for the legal word embedding modelling and the evaluation benchmark dataset. The process can be divided into three major steps. The first is to collect the textual data of the codex articles, after completing pre-processing such as data cleaning and word segmentation. Then the data is converted into a dataset whose format can be used by subsequent tools. Next, we set various hyper parameters to adjust the word embedding training tool and then import the dataset into the tool for training. The third step is having legal experts provide various legal relationships and corresponding vocabulary references from codices, using TensorBoard to assist the legal experts in reviewing the vocabulary in the word embedding models, and, finally, establishing the LARQS evaluation dataset manually. When using TensorBoard and identifying Chinese lexical words inappropriate due to word segmentation, we use the dotted line in Figure 2 to return to the pre-processing step to manually fine-grain the word segmentation, and then go back to the second step to re-train the word embedding model.

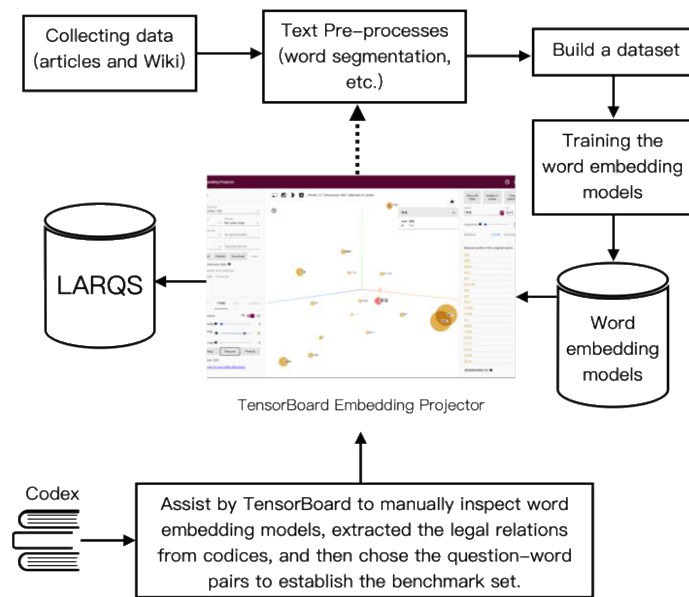


Figure 2. The pipeline of word embedding modeling and benchmark dataset construction.

4. CATEGORIES OF LEGAL TERM RELATIONS

Through expert manual inspection, we extracted five categories of legal relations from 2,388 codices to experiment with analogical reasoning. We then established a dataset consisting of 1,134 legal analogical reasoning questions-words (Legal Analogical Reasoning Question Set, LARQS). The five categories of legal term relations included “Prime and Deputy”, “Rights and obligations”, “Execute and staff”, “Operation”, and “Rights”. Table 2 is a sample of this.

Table 2. A sample of Legal Analogical Reasoning Question Set

Relation Category	Example	Questions	Words
Prime and Deputy (正副關係)	正首長(chief), 副首長(deputy chief) 總統(president), 副總統 (vice president)	870	30
Rights and obligations (權利義務相對人)	債權人(creditor), 債務人(debtor) 典權人 (dien-holder), 出典人(dian-maker)	42	7
Execute and staff (執業行為與人員)	查核(audit), 會計師(certified public accountant) 競選(campaign,), 候選人(candidate)	42	7
Operation (客體操作行為)	疫苗(vaccine), 接種 (vaccination) 森林(forest), 墾殖(cultivating)	90	10
Rights (權利與權利人)	債權(claims), 債權人(creditor) 電路布局權(circuit layout rights), 電路布局權人 (circuit layout right owner)	90	10

After estimating the accuracy of these two embedding models (the legal word embedding model and the Chinese Wikipedia word embedding model) with LARQS, we found that the various legal relations listed in Table 2 are common in the legal word embedding model and that legal relations exist in the general word embedding model too. This finding is presented in this paper and will be discussed in the next section.

5. EXPERIMENTAL RESULTS

We use the accuracy function by gensim provided and various benchmarks sets in this paper. The accuracy of the codex word embedding model, established in skip-gram, and the experimental results from evaluating with the LARQS benchmark dataset are exhibited in Figure 3. In this same figure, the size of the word embedding affects accuracy, while iteration times are an additionally important parameter. In this experiment, the number of iterations range from 100 to 200, the word embedding size ranges from 700 to 850 dimensions, and the average performance accuracy is as high as 65%. However, when the iterations were set to 1000, the accuracy proved inferior to the iterations being set to 100.

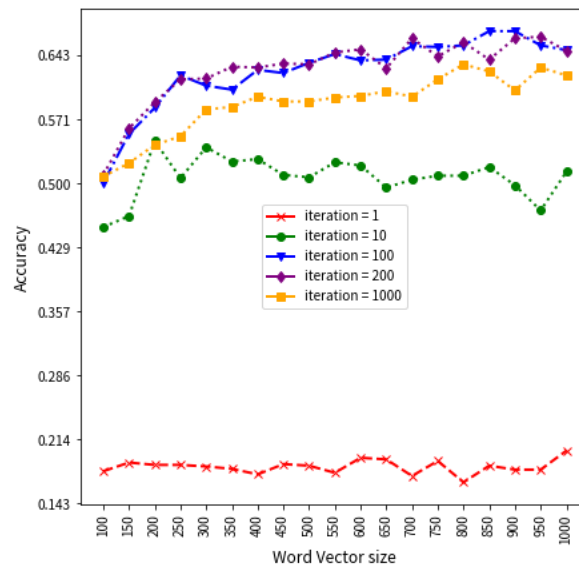


Figure 3. Accuracy affected by word embedding size and training iterations (skip-gram)

The word embedding model proposed by Mikolov has another architecture, CBoW. This paper uses the CBoW architecture to establish a word embedding model for the same codex dataset and then uses this paper's LARQS evaluation benchmark dataset to evaluate accuracy. The average accuracy is not superior to that of Skip-gram architecture, as shown in Figure 4. Therefore, the subsequent experiments in this paper use the Skip-gram architecture word embedding model to evaluate accuracy.

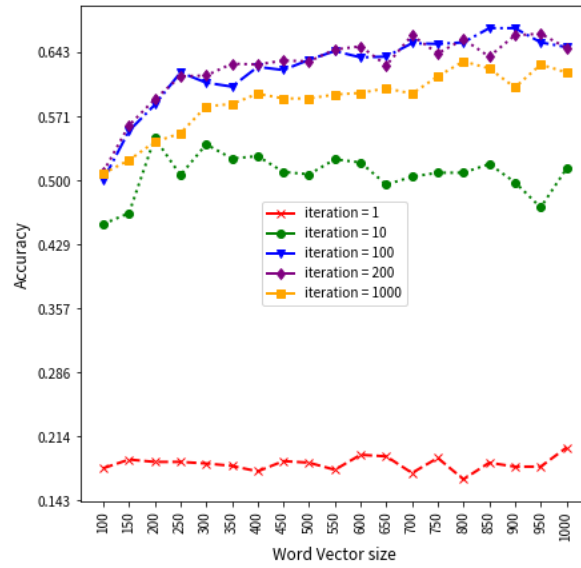


Figure 4. Accuracy affected by word embedding size and training iterations (CBoW)

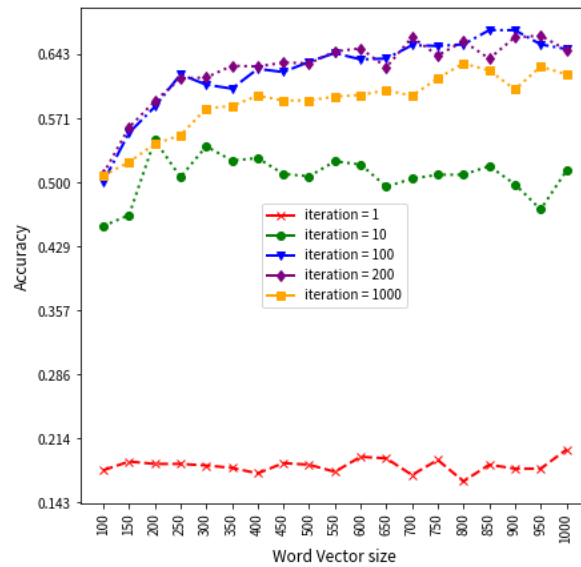


Figure 5. Accuracy affected by word embedding size and training iterations (zh-tw Google set)

To observe the performance differential between the LARQS and other public benchmark datasets, we first selected the evaluation benchmark dataset released by Google and translated by CKIP Group. As can be seen from the experimental results in Figure 5, when the Google evaluation benchmark dataset translated by CKIP Group is subjected to different training iterations, the accuracy of the Chinese Codex word embedding model is not outstanding.

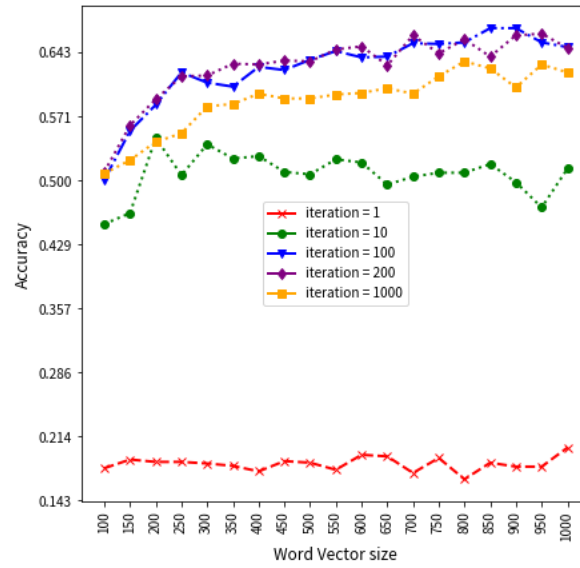


Figure 6. Accuracy affected by word embedding size and training iterations (CA8/morphological)

Figures 6 and 7 present our evaluation of our Chinese codex word embedding model with the CA8 evaluation benchmark datasets published by Shen Li [14]. The CA8 evaluation benchmark datasets fall into two categories: morphological and semantic. Figure 6 shows the experimental results of evaluating the accuracy of the Chinese Codex word embedding model with the CA8 morphological evaluation benchmark dataset. The performance of the Chinese codex word embedding model is very low. Figure 7 is the CA8 semantic benchmark dataset, which is used to evaluate the accuracy of the Chinese Codex word embedding model and whose performance is similarly less than ideal.

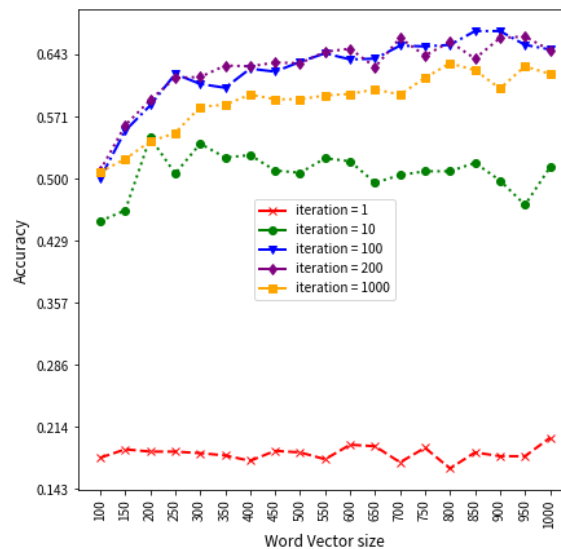


Figure 7. Accuracy affected by word embedding size and training iterations (CA8/semantic)

Figure 8 summarizes the experimental results of evaluating the aforementioned datasets. Each of the previous benchmark dataset experiments using the word embedding model established by 2,388 codices is collected in this paper. The size of the word embedding models ranged from 100 to 1000, with iterations numbering 100. Next, the simple algebraic calculation method that Mikolov [7] proposed to evaluate accuracy is applied. As Figure 8 reveals, the top accuracy achieved with LARQS in this paper is 67.02% using word embedding in 850 dimensions. This significantly outperforms the linguistic-based CA8 evaluation benchmark dataset by 8.2%, and also improves upon Google's evaluation benchmark dataset translated by the CKIP Group by some 27.58%. Table 3 illustrates the accuracy achieved with different word embedding sizes in the same Chinese Codex word embedding model with regard to each evaluation dataset.

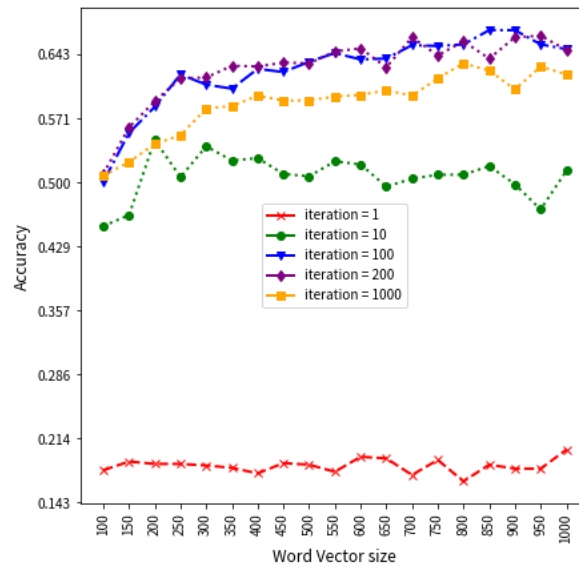


Figure 8. Estimating accuracy of codex word embedding model with different benchmark

Table 3. The accuracy achieved with different word embedding sizes in the Chinese Codex word embedding model with regard to each benchmark dataset.

Benchmark Dimension	LARQS	Google Set	CA8/morphological	CA8/semantic
100	0.5	0.2758	0.1064	0.0285
200	0.5846	0.2758	0.1244	0.0464
300	0.6093	0.2758	0.1265	0.0678
400	0.6269	0.2758	0.1164	0.0714
500	0.6349	0.2758	0.1184	0.075
600	0.6375	0.2758	0.1204	0.0714
700	0.6534	0.2758	0.1204	0.075
800	0.6543	0.2758	0.1184	0.0785
850	0.6702	0.2758	0.1205	0.0679
900	0.6702	0.2758	0.1164	0.075
1000	0.649	0.2758	0.1084	0.0821

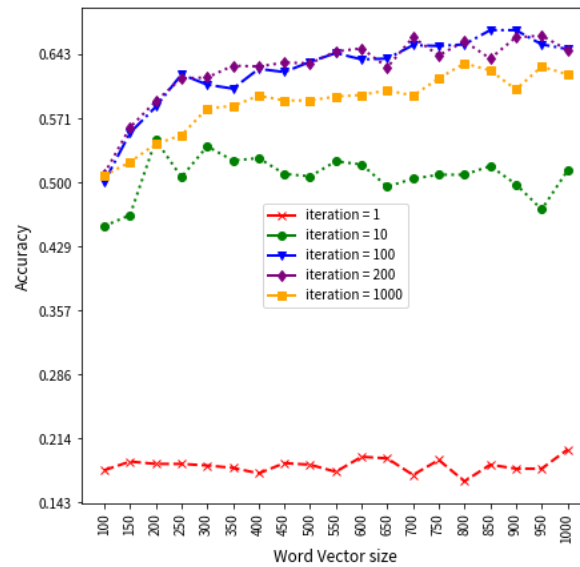


Figure 9. Estimating accuracy of Wikipedia (zh-tw) word embedding model with different benchmark

To explore the legal relationship between the words in the word embedding model, we also collected the Chinese version of Wikipedia, which covers a wide range of content, as a dataset. This experiment builds 19 differently-sized word embedding models based on the Chinese version of Wikipedia up to December 1, 2019. The word embedding size ranges from 100 to 1000, with 50 dimensions as the interval range for each model. Next, applying the LARQS in this paper, the Google benchmark set translated by Academia Sinica and the two CA8 benchmark sets (translated from simplified to traditional Chinese) are used to estimate the accuracy of the Wikipedia word embedding model. Figure 9 shows the accuracy of the Wikipedia word embedding model evaluated using different benchmark sets. In the best-case scenario for LARQS, i.e., when the word embedding size is 750, an accuracy level of 65.24% can be achieved. This is higher than the translated Google evaluation data set whose peak accuracy is 55.59% when the word embedding size is 400. Table 4 illustrates the accuracy of various word embedding sizes of the Wikipedia Chinese word embedding model, estimated using different benchmark datasets. Reviewing the results of this experiment (even for general documents, and estimating the word embedding model established with the LARSQ dataset for legal relations), when compared with the 14 categories in the benchmark dataset released by Google (i.e., in linguistics, capital-common-countries, currency-names-and-countries, and using the CA8 benchmark dataset based purely on linguistics), the LARQS dataset in this paper can better demonstrate the universality of legal relations between vocabulary relations in the word embedding model.

Table 4. The accuracy of various word embedding sizes of the Wikipedia Chinese word embedding model, estimated using different benchmark datasets.

Benchmark Dimension	LARQS	Google Set	CA8/morphological	CA8/semantic
100	0.5372	0.4204	0.2278	0.3554
200	0.5993	0.5013	0.2450	0.4172
300	0.6277	0.5249	0.2474	0.4374
400	0.6401	0.5559	0.2464	0.4557
500	0.6348	0.5488	0.2355	0.4609
600	0.6436	0.5482	0.2063	0.4501
700	0.6454	0.5423	0.2076	0.4465
750	0.6525	0.5427	0.1994	0.4488
800	0.6401	0.5233	0.1914	0.4392
900	0.6383	0.5378	0.1996	0.4608
1000	0.6294	0.5373	0.1886	0.4509

6. FUTURE WORK AND CONCLUSION

Applying simple algebraic calculations to obtain the deviation values between word vectors, combined with manually selected Chinese words (vocabulary), and mining the relationship between the vocabulary in the word embedding model with its legal relationship, is a paradigm of especial interest proposed in this paper. Our experiment's results indicate that legal relations might be ubiquitous in the word embedding model based on legal provisions, aka codices, and word embedding models covering a wider range of content. We are also publishing the LARQS benchmark dataset and the Chinese codex word embedding model from our experiment to expedite carrying out NLP tasks related to Chinese law. In the future, we hope that automated methods will be developed to unearth additional legal relationships and enrich the LARQS dataset proposed in this paper. These, applied with this experiment's word embedding model, combined with more advanced machine learning algorithms, could be applied to excellent effect on more complex NLP tasks such as document generation, automatic contract review, document classification, and machine translation.

REFERENCES

- [1] Yu, L.-C., et al., (2017) "Refining word embeddings using intensity scores for sentiment analysis", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 26, No. 3, pp 671-681.
- [2] Ge, L. and T.-S. Moh, (2017) "Improving text classification with word embedding", *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, pp 1796-1805
- [3] Choi, H., K. Cho, and Y. Bengio, (2017), "Context-dependent word representation for neural machine translation", *Computer Speech & Language*, Vol. 45, pp 149-160.
- [4] Zhang, B., et al., (2017), "A context-aware recurrent encoder for neural machine translation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25, No. 12, pp 2424-2432.
- [5] Bin Wang, et al., (2019) "Evaluating word embedding models: Methods and experimental results", *APSIPA transactions on signal and information processing*, Vol. 8
- [6] Bakarov, A., (2018) "A survey of word embeddings evaluation methods", *arXiv preprint arXiv:1801.09536*
- [7] Tomas Mikolov, et al., (2013a) "Efficient estimation of word representations in vector space", *arXiv preprint, arXiv:1301.3781*
- [8] Tomas Mikolov, et al., (2013b) "Distributed representations of words and phrases and their compositionality", *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol. 2, Curran Associates Inc., Lake Tahoe, Nevada, pp 3111-3119.

- [9] Muneeb, T.H., S. Sahu, and A. Anand, (2015) "Evaluating distributed word representations for capturing semantics of biomedical concepts", *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015)*, pp 158-163
- [10] Nikfarjam, A., et al., (2015) "Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features", *Journal of the American Medical Informatics Association*, Vol. 22, No. 3, pp 671-681.
- [11] Soricut, R. and F. Och, (2015) "Unsupervised Morphology Induction Using Word Embeddings", *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*. Denver, Colorado, Association for Computational Linguistics, pp1627-1637
- [12] Gladkova, A., A. Drozd, and S. Matsuoka, (2016) "Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't", *Proceedings of the NAACL Student Research Workshop*, pp 8-15.
- [13] Köper, M., C. Scheible, and S.S. im Walde. (2015) "Multilingual reliability and 'semantic' structure of continuous word spaces", *Proceedings of the 11th international conference on computational semantics*, pp 40-45.
- [14] Li, S., et al., (2018) "Analogical reasoning on Chinese morphological and semantic relations", *arXiv preprint*, arXiv:1805.06504
- [15] Xue, N. (2003), "Chinese word segmentation as character tagging", *International Journal of Computational Linguistics & Chinese Language Processing, Special Issue on Word Formation and Chinese Language Processing*, Vol. 8, No. 1, pp 29-48.
- [16] Chang, P.-C., M. Galley, and C.D., (2008) "Manning, Optimizing Chinese word segmentation for machine translation performance", *Proceedings of the Third Workshop on Statistical Machine Translation*, Association for Computational Linguistics, Columbus, Ohio, pp 224-232.
- [17] Hai, Z., et al., (2019) "Chinese Word Segmentation: Another Decade Review (2007-2017)", *arXiv*, abs/1901.06079.
- [18] Sproat, R. and C. Shih, (1990) "A statistical method for finding word boundaries in Chinese text", *Computer Processing of Chinese and Oriental Languages*, Vol. 4, No. 4, pp 336-351.
- [19] Fu, J., et al., (2020) "RethinkCWS: Is Chinese Word Segmentation a Solved Task?", *arXiv preprint*, arXiv:2011.06858.
- [20] Xie, Z., (2017) "Closed-Set Chinese Word Segmentation Based on Convolutional Neural Network Model", *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Cham, Springer International Publishing, pp 24-36
- [21] Chen, X., et al., (2015) "Long short-term memory neural networks for Chinese word segmentation", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp 1197-1206
- [22] Bengio, Y., et al., (2003) "A neural probabilistic language model", *The Journal of Machine Learning Research*, Vol. 3, pp 1137-1155.
- [23] Dumais, S.T., et al., (1988) "Using latent semantic analysis to improve access to textual information", *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp 281-285
- [24] Blei, D.M., A.Y. Ng, and M.I. Jordan, (2003) "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, Vol. 3, pp 993-1022.
- [25] Pennington, J., R. Socher, and C.D. Manning, (2014) "Glove: Global vectors for word representation", *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp 1532-1543.
- [26] Vulić, I. and N. Mrkšić, (2017) "Specialising word vectors for lexical entailment", *arXiv preprint*, arXiv:1710.06371.
- [27] Vylomova, E., et al., (2016) "Take and Took, Gaggles and Geese, Book and Read: Evaluating the Utility of Vector Differences for Lexical Relation Learning", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, Association for Computational Linguistics, Vol. 1, pp 1671-1682
- [28] Camacho-Collados, J., L. Espinosa-Anke, and S. Schockaert, (2019) "Relational word embeddings", *arXiv preprint* arXiv:1906.01373.
- [29] Bouraoui, Z., J. Camacho-Collados, and S. Schockaert, (2020) "Inducing relational knowledge from BERT", *AAAI Conference on Artificial Intelligence*, Vol. 34, No. 5, pp 7456-7463.
- [30] Sun, Y., et al., (2014) "Radical-Enhanced Chinese Character Embedding", *Neural Information Processing*, Cham, Springer International Publishing, pp 279-286

- [31] Li, Y., et al., (2015) “Component-Enhanced Chinese Character Embeddings”, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Association for Computational Linguistics, pp 829-834.
- [32] Yin, R., et al., (2016) “Multi-Granularity Chinese Word Embedding”, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, Association for Computational Linguistics, pp 981-986.
- [33] Kang, R., et al., (2019) “Learning Chinese Word Embeddings With Words and Subcharacter N-Grams”, *IEEE Access*, Vol. 7, pp 42987-42992.
- [34] Yu, J., et al., (2017) “Joint Embeddings of Chinese Words, Characters, and Fine-grained Subcharacter Components”, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Association for Computational Linguistics, pp 286-291.
- [35] Su, T.-R. and H.-Y. Lee, (2017) “Learning chinese word representations from glyphs of characters”, *arXiv preprint*, arXiv:1708.04755.
- [36] Cao, S., et al., (2018) “cw2vec: Learning Chinese word embeddings with stroke n-gram information”, *Thirty-second AAAI conference on artificial intelligence*, pp 5053-5061.
- [37] Yang, Q., et al., (2021) “Pronunciation-Enhanced Chinese Word Embedding”, *Cognitive Computation*, Vol. 13, No. 3, pp 688-697.
- [38] Chen, C.-Y. and W.-Y. Ma, (2018) “Word Embedding Evaluation Datasets and Wikipedia Title Embedding for Chinese”, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pp 825-831.
- [39] Turney, P.D., (2008) “A uniform approach to analogies, synonyms, antonyms, and associations”, *arXiv preprint*, arXiv:0809.0124.
- [40] Turney, P.D. and P. Pantel, (2010) “From frequency to meaning: Vector space models of semantics”, *Journal of artificial intelligence research*, Vol. 37, pp 141-188.
- [41] Aleksandr Drozd, Anna Gladkova, and S. Matsuoka, (2016) “Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen”, *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*, pp 3519-3530.
- [42] Levy, O. and Y. Goldberg, (2014) “Neural word embedding as implicit matrix factorization”, *Advances in neural information processing systems*, Vol. 27, pp 2177-2185.
- [43] Levy, O., Y. Goldberg, and I. Dagan, (2015) “Improving distributional similarity with lessons learned from word embeddings”, *Transactions of the association for computational linguistics*, Vol. 3, pp 211-225.
- [44] Ash, E., & Chen, D. L. (2017) “Judge embeddings: Toward vector representations of legal belief”, *Technical report*. https://users.nber.org/~dlchen/papers/Judge_Embeddings_slides.pdf

AUTHORS

Chun-Hsien Lin is currently a Prosecutorial Affairs Officer of Taiwan High Prosecutors Office and pursuing the Ph.D. degree from the National Taiwan University of the Department of Computer Science & Information Engineering. His research is mainly on AI and law, natural language processing, machine learning, and so on.



Pu-Jen Cheng received his Ph.D. in Computer Science at National Chiao Tung University in 2001. He went to the Institute of Information Science, Academia Sinica, as a Postdoctoral Fellow for more than four years. Starting from August 2006, he joined the department of Computer Science and Information Engineering faculty at the National Taiwan University, and also jointly appointed at the Graduate Institute of Networking and Multimedia, National Taiwan University. He is a member of the ROC Phi Tau Phi Scholastic Honor Society and the ACM/SIGIR.



AN IR-BASED QA SYSTEM FOR IMPACT OF SOCIAL DETERMINANTS OF HEALTH ON COVID-19

Priyanka Addagudi and Wendy MacCaull

Department of Computer Science, St. Francis Xavier University, Canada

ABSTRACT

Question Answering (QA), a branch of Natural Language Processing (NLP), automates information retrieval of answers to natural language questions from databases or documents without human intervention. Motivated by the COVID-19 pandemic and the increasing awareness of Social Determinants of Health (SDoH), we built a prototype QA system that combines NLP, semantics, and IR systems with the focus on SDoH and COVID-19. Our goal was to demonstrate how such technologies could be leveraged to allow decision-makers to retrieve answers to queries from very large databases of documents. We used documents from CORD-19 and PubMed datasets, merged the COVID-19 (CODO) ontology with published ontologies for homelessness and gender, and used the mean average precision metric to evaluate the system. Given the interdisciplinary nature of this research, we provide details of the methodologies used. We anticipate that QA systems can play a significant role in providing information leading to improved health outcomes.

KEYWORDS

Question Answering, Ontology, Information Retrieval, Social Determinants of Health, COVID-19.

1. INTRODUCTION

The world wide web has allowed researchers and decision-makers in medical and other domains to easily access data and documents and build vast repositories of related knowledge. However, determining the relevant information for a particular research problem or decision-maker is difficult. The increased power to perform computations has enabled the application of Artificial Intelligence techniques (NLP based QA systems [8], Machine Learning, etc.) to the problem of finding relevant information.

Coronavirus infection emerged in December 2019 in Wuhan city, China. The infection rapidly spread across parts of China and later worldwide. The COVID-19 pandemic is a worldwide crisis endangering the health of everyone on the planet. Droplets from mouth, nose and direct contact with a person infected with COVID leads to transmission of the virus. The Social Determinants of Health (SDoH) are socio-economic conditions that impact people's health. The authors of paper [11] discuss how the SDoH impact disadvantaged populations during times of crisis. Studying SDoH and how they impact crises like COVID can help decision-makers manage health emergencies so that everyone has an equal opportunity to stay healthy. A study done on impacts of SDoH on COVID-19 by [9] states preliminary evidence from surveillance and media reports have shown that SDoH contributes to high rates of COVID-19 infection, hospitalization, and

mortality. Recently, the CTV news article [10] stated the socio-economic and health inequities are the topmost important factors out of five big lessons learned from the pandemic. Considering the importance of COVID-19 related topics, the Allen Institute for AI, in collaboration with other organisations and Kaggle, released the COVID-19 Open Research Dataset (CORD-19) [50]. The scientific papers on Covid-19 and related historical coronavirus research are growing, and CORD-19 is the resource for all of them. This dataset aims to enable researchers to work with new tools to analyse and overcome the problems associated with coronavirus. Overall evidence indicating a crucial role for SDoH and related social factors in shaping health has become so compelling that it cannot be ignored [57].

The objective of this work is to develop an integrated architecture (Domain SDoH and QA system) for a Question and Answering system dealing with a corpus related to “Impact of SDoH and associated risk factors on transmission and outcomes of COVID-19”. The NLP based QA system allows the users to input the query in natural language, which reduces the technicality and time needed to get the information. According to our literature search, there is no QA system for the impact of SDoH factors on COVID-19. This has motivated the long-term goal of our research which is to create a QA system that can assist people in understanding the impact of SDoH on COVID-19 or future pandemics due to infectious diseases.

Question answering (QA), is a branch of information retrieval and natural language processing (NLP) [1]. QA systems automatically provide answers to questions posed in a natural language. They are different from Information Retrieval (IR) systems or search engines like Google that return a ranked list of relevant sources based on a set of keywords. A QA system finds and returns relatively short and concrete answers in the form of: a sentence, a paragraph, a fragment of the text, or even a word that answers a given question, by analysing many documents where the answers may be found. QA systems can reduce technical difficulties by enabling humans to interact with machines using natural languages instead of programming languages or text-based commands. QA systems are evolving worldwide and used in a wide range of application areas, from biomedicine to tourism [2]. Several literature reviews have focused on a variety of aspects of QA systems, e.g., domain [3][4], information retrieval paradigm [5], hybrid-based paradigm [6]. However, no established relationships exist between domains, algorithms, techniques, and systems [7].

This paper showcases our prototype QA architecture, AQuA, demonstrating how a QA system integrated with a domain, can answer questions related to severe outcomes of COVID-19 due to some SDoH factors (homelessness and gender). The QA system developed allow users (decision-makers) to understand and quickly determine some circumstances (related to SDoH) that can lead to transmission and severe outcomes associated with the Covid-19 infection. The motivating research for this work is MEANS, a QA system built to address the problems in the medical domain combining NLP techniques and semantic web technologies [12]. Our architecture differs from MEANS in various factors, namely: a) Named Entity Recognition (NER) and Relation Extraction (RE) processes are replaced with rule-based triple extraction compared to that work. b) Three layers for similarity measures are used for linking. c) The Resource Descriptive Framework (RDF) triple store is replaced by Elastic Search database, questions are converted to Elastic Search queries and the Elastic Search scoring model is used for answer ranking. We built an ontology for the two SDoH factors under consideration in this work (i.e., homelessness and gender) and COVID-19 to support the QA system.

The remainder of this paper is described as follows: section 2 outlines the background required for this research and related work, section 3 details the architecture and implementation, section 4 describes the evaluation results, section 5 concludes this work and outlines some future works.

2. BACKGROUND AND RELATED WORK

2.1. Background

NLP is a field that deals with the interactions between computers and humans, especially how to program computers to process and analyse large amounts of natural language data [13]. QA is a Computer Science discipline within the field of NLP, which is concerned with building systems that can automatically answer questions posed in a natural language [14]. The three most common essential modules in most QA systems are document retrieval module, question processing module, and answer extraction and formulation module. As stated in [15], QA systems are different from IR systems (aka, search systems) which allow users to input keywords to search and returns lists of documents in response. The difference between IR and QA systems is that QA systems internally have a stronger dependency on NLP techniques such as parsing, named-entity detection, semantic role labelling, tree-matching, and, in some implementations, logical inference. Any QA system typically has many modules, and its performance varies and depends on how the components are integrated within an algorithm. There are two types of QA systems: a) Open-domain question answering deals with everything (not specific to any domain) so there are many documents to retrieve answers from. b) Closed-domain question answering deals with questions on a specific domain, so there are a limited number of documents to retrieve answers from. Closed-domain QA has some advantages over open-domain QA, e.g., users have an idea about what domain they are using the system for and do not use it for general purposes. Also, the users can leverage domain properties to build vocabularies, ontologies and other models.

2.1.1. Components in NLP pipeline

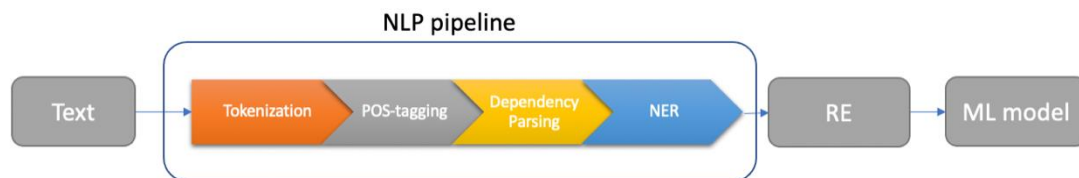


Figure 1: NLP pipeline

The NLP pipeline is shown in Figure 1. Tokenization is the task that splits the sentences in documents into sentence tokens or words in a sentence to meaningful word tokens. The tokens are identified and matched with the training data used for other processes in the pipeline. POS-tagging refers to assigning every word in a sentence with its part of speech (noun, verb, adverb, and so on) [16]. In general, Hidden Markov models [17] are used for POS-tagging task because information about the context of a word can be used to predict what the POS-tag might be. There are also rule-based POS-taggers, which use predefined rules to perform the tagging or learn rules from the corpus and rely less on probabilistic methods. Named entity recognition seeks to locate and classify entities mentioned in the text into pre-defined categories like chemical, symptom, disease etc. A named entity is a real-world object usually with a proper name - examples are Donald Trump, Italy, Facebook etc. Dependency Parsing is a way to analyse a sentence or break up a sentence to understand the structure of a sentence. Parsing is a technique that can be applied to any statement with formal grammar. RE is the task of extracting semantic relationships from text (usually between two entities) [18]. This cleaned and pre-processed data from the pipeline is converted to vectors and used to train the ML models.

2.1.2. Semantic Web Technologies

Ontology (a branch of philosophy known as metaphysics) is the study of the nature of being, becoming, existence, or reality [19]. It deals with questions concerning what things exist or can be said to exist and how they can be grouped or classified based on similarities and differences. Ontologies, which are richer than terminologies [20] can be used to formally represent knowledge within a domain as a set of concepts, relations between concepts, instances of the concepts [21] and axioms. A pair of concepts or instances with a relation between them is known as 'semantic triple'. One semantic triple can be connected to another semantic triple whenever they share any common concept. In knowledge graphs, which are frequently used to represent ontologies, these concepts or instances aka entities are represented as nodes and the relations are represented as edges. Each ontology class and relation may have a Uniform Resource Identifier (URI) associated with it which is used to identify the concepts and relations. Resource Description Framework (RDF) expresses information about resources like people, physical objects, and abstract concepts [22]. RDF facilitates sharing of information on the web between different applications without loss of its meaning by providing a common framework for expressing the information. The attractive feature of RDF is the ability to distribute the development of an ontology (defining a set of data and their structure for other programs to use them) by basing terms in dereference able, globally unambiguous identifiers. To achieve this, RDF uses IRI (International Resource Identifier), which is a generalization of URIs whose scalability has been proven by the success of the web [23]. This generalization allows non-ASCII characters to be used in IRI character strings [24]. The Universal Resource Locator is one form of IRI. Another form of IRI provides an identifier for the resource but does not provide its location or explain how to access it. Triples in RDF are expressed in statements composed of a subject, object, and a predicate (i.e., relationship) between them.

2.1.3. Elastic Search

Search over RDF data is supported by two methods (a) By translating keyword queries to structured (SPARQL) queries [25] (b) By building or leveraging Information Retrieval Systems using classical IR methods for indexing and retrieval [27]. Our architecture is based on method (b) involving ElasticSearch, so the results or scoring/ranking is dependent on commonly used IR ranking functions.

ElasticSearch is open-source and one of the most popular search engines under the Apache 2.0 license [26]. ElasticSearch allows storage and searches a large amount of data offering a distributed architecture and emphasises scalability and reliability. ElasticSearch has a powerful query language known as Domain Specific Language (DSL), which supports advanced search features [27] such as search based on filter-context and query-context. Apache Lucene library can handle all types of data (textual, numerical, geospatial, structured, and unstructured), allowing the users of ElasticSearch were able to focus on scalability, usability, and performance. ElasticSearch provides an easy way to index documents enabling users to quickly query the nearest neighbour using a similarity metric based on TF-IDF [28]. Retrieval accuracy is affected by indexing and retrieval options in ElasticSearch. In ElasticSearch, the default model for similarity matching is BM25 [28] based on TF-IDF similarity measure (described in section 2.1.4). The data in ElasticSearch is stored in indices containing a different type of documents. This data stored can be searched and updated by ElasticSearch.

For indexing in ElasticSearch, each term will be used as a key to a multi-value map creating key-value pairs, which is called an 'inverted index'. The retrieval process depends on the i) query type, ii) weighing methods and iii) similarity models offered by ElasticSearch. There are two types of query clauses: a) filter-context - using exact match with query; b) query-context - using relevance

scoring. As our interest is primarily free text search, we focus on query-context. Queries are further categorised into single-match queries, multi-match queries and boolean queries. As the name suggests, a single-match query is executed over a single field in a database, while a multi-match query is executed over multiple fields in a database. A Boolean query is the combination of single and multi-match query types. Boolean queries use clauses such as MUST or SHOULD. Weighing is an important factor for improving relevance at retrieval time. In this method, we usually apply weights on various fields. For instance, in ElasticSearch the field containing the object keyword is twice as important as the fields containing the subject and predicate keyword and has more weight. The way scoring works in real-time is different from generic scenarios. E.g., (as found in [29]) a document titled *Shard Selection Algorithm* contains the term algorithm; however, it is not relevant to the query *Algorithm for pathfinding*. Determining the relevancy to a query is a fundamentally hard problem. We can overcome this problem by implementing some of the most common scoring models like the TF-IDF scoring model.

The RDF datasets (aka web-of-data) are generally queried through a structured query language (such as SPARQL). But this is a difficult task for any person who is not proficient in SPARQL even though they are familiar with keyword search. Hence, there is a need for an effective method for keyword search over RDF datasets. Keywords are the phrases that a user type into search engines to retrieve an answer. Keyword search is used to optimise the search engine by adding weights to the keywords. In the example *Shard Selection Algorithm*, word *Shard* is given more weightage (using TF-IDF scoring model) than other two words so that search engine recognises the keyword for the search is *Shard*. There is widespread evidence [27] to the use of out-of-the-box IR systems like ElasticSearch in many contexts. The experiment by [27] investigated how such existing document-centric Information Retrieval Systems (IRS) can be used for enabling keyword search over RDF datasets, and how they perform compared to dedicated keyword search systems for RDF.

2.1.4. Linking

Linking is a process of mapping the extracted entities (subject or object) and relations (predicates) from the triple extraction process to the concepts and relations present in the knowledge bases (ontologies). In general, concepts or classes and relations in the knowledge base have unique identifiers. The combination of NLP and Semantic web technologies enables users to combine both structured and unstructured data. The linking process is difficult due to the high ambiguity of entity mentions and relations, which includes polysemy and multiword synonym [30]. Although this method is difficult, it can help search engines to disambiguate and retrieve the closest answer as the top ranked answer.

Linking can be performed in many methods [30]: a) Methods based on similarity - this can be done using Fuzzy matching, TF/IDF, Cosine similarity, and others. b) Methods based on machine learning - this can be done using classifiers like Support Vector Machine (SVM) classifiers and others. c) Methods based on graphs - this can be done by constructing a graph with nodes that are entities and edges which are relations. There are 2 main kinds of similarity measures, syntactic and semantic. Fuzzy matching is syntactic, cosine similarity is semantic; TF-IDF, N-gram are hybrid. Fuzzy logic helps in dealing with the problem of knowledge representation in an environment of uncertainty and imprecision. Fuzzy matching is a string-matching algorithm that measures the similarity between two strings using edit distance, also known as Levenshtein edit distance [31].

TF-IDF is also known as Term Frequency - Inverse Document Frequency [29]. TF-IDF can be used to calculate the similarity between two records by considering the frequency of the word in the data or documents. The first term in TF-IDF is Term Frequency, $Tf_{(t, d)}$ which means the

frequency of the word t in the document d , calculated as Equation (1). Here, it assumes that a document having more than one match of the highest weighted term is a better match than a document having a single match. Document Frequency of term t , D_{ft} , over all documents is calculated by counting the number of postings for a term in the inverted index, and the Inverse Document Frequency (IDF) for a term t in document d , Idf_t is calculated as Equation (2).

$$T f_{(t, d)} = \sqrt{f_t} \quad (1)$$

$$Idf_t = \log\left(\frac{N}{D_{ft} + 1}\right) \quad (2)$$

where N is the total number of documents. For scoring, the $T f_{(t, d)}$ weights are combined with Idf_t weights to multiplied by the score, as shown in Equation (3).

$$\text{score}(d, q) = \sum_{t \in q} T f_{(t, d)} \times Idf_t \quad (3)$$

where q is the query. Soft TF-IDF is very similar to the TF-IDF concept which considers how frequently various combinations of words appear in data and calculated as Equation (4):

$$\text{Sim}_{\text{SoftTFIDF}}(x, y) = \sum_{\omega \in \text{CLOSE}(\theta, x, y)} \frac{(V(\omega, x))}{\sqrt{\sum_{\omega} V(\omega, x)^2}} \frac{(V(\omega, y))}{\sqrt{\sum_{\omega} V(\omega, y)^2}} D(\omega, y) \quad (4)$$

Where x , y are two strings from documents X , Y , ω is the word in the string and θ is the threshold defined for comparison, $V(\omega, x)$ and $V(\omega, y)$ are defined as the TF-IDF weight of token ω in string x and y given by frequency count in X and Y , $D(\omega, y)$ is essentially a normalizing coefficient, which dampens the impact on the Soft TF-IDF [67].

Context can be very important when working with textual data. We may sometimes lose the context in vector representations of a word, knowing only the count of every word. To some extent, N-grams, and in particular bi-grams, will help us solve this problem. An n-gram is a string of elements such as letters, words etc., that appear in a continuous sequence. We will be dealing with words being the item, but based on the use case, it could be letters, syllables, or sometimes, in the case of speech, phonemes. When $n=2$, it is called a bi-gram. Bi-grams in a text can be calculated using the conditional probability of a token with respect to its preceding token. Another way of calculating bi-grams is by choosing words that appear next to each other, but it is more effective to use bi-grams that are likely (using the conditional probability) to appear as a pair; such a bi-gram is called a collocation. Character n-gram (charNgram) is a character-based compositional model which is used to embed textual sequences [32]. In this method, each word is represented as a bag of character n-grams. The vector representation of a string or word is associated with each character n-gram. The end character embedding is the average of the distinct character n-gram embeddings. Using the character embeddings methodically and efficiently provides morphological features [33].

The model implemented in this work compares the scores from all the 3 models and finds the similarity between entities with ontology concepts. By using all three methods, we can deal with various categories of terms like: misspelled words, words in a similar context, large vocabularies, and many rare words.

2.1.5. Domain – SdoH

Existing social inequities in health may increase the risk of severe COVID-19 outcomes, such as hospitalization and death [9][10][11]. Racialized populations, Socioeconomic status, Homeless

populations, Gender, Incarcerated populations, Education are some SDoH factors. Obesity, hypertension, diabetes, cardiovascular disease, and chronic respiratory disease and asthma are a few risk factors or conditions that may lead to co morbidities (overlap of different conditions) that may be associated with increased risk for severe outcomes from COVID-19. To help vulnerable people/groups in times of emergency, the focus should be on the roots of the problem. There is a need for all, not just decision-makers, to understand how social determinants and associated risk factors can impact the mortality rate as low mortality rates are associated with community support and cohesion. As stated in [11], pandemics are more of a social problem than a healthcare problem. Understanding factors like SDoH that play an important role in health and healthcare can facilitate access to medical and non-medical needs for everyone on a more equitable basis. Integration of SDOH into efforts to eliminate disparities in health and healthcare can be one solution to reduce the impact (transmission and outcomes) globally [11].

2.2. Related Work

BASEBALL [34] and LUNAR [35] are some of the oldest and well know QA systems that answer questions related to the US baseball league and geological analysis of rocks returned by the Apollo moon mission, respectively. The most noteworthy system is IBM Watson [36], which is the best example of a successful QA system. Other commercial products in the area of personal assistants include Apple's Siri (in 2011), Amazon's Alexa (in 2014) and Microsoft's Cortana (in 2014), Samsung's Bixby, and Google Assistant. The adoption of QA-based personal assistants (capable of answering variety of questions) has been observed over the last few years.

The QA Systems architecture depends on the underlying knowledge source like plain text, data graphs (RDF), or mixed (plain text and data graphs). The authors of [1] have done a comprehensive survey on the question answering system over RDF and Linked Data, documents, and mixtures of these to get a clear understanding of the QA systems. An RDF KB describes real-world entities as well as the relations between them. Integration of RDF KBs with other KBs or data sources (e.g., CSV, SQL tables) is necessary as it makes KBs able to support QA systems both in Open and Closed domains. Availability of tools or techniques relative to the integration of various databases makes the integration task easier. This data integration task can be done in many ways; some are: 1) interlinking different KBs based on ontology mapping methodologies [37] [38], 2) transforming and properly integrating other data source types into RDF ontologies using semantic labelling techniques [39] [40].

As our method is based on keyword search over RDF data by adapting an IR system using classical IR methods for indexing and retrieval, we will report related work to showcase the difference in our approach. The paper [27] presented a study that investigates the challenges and techniques to overcome them while querying RDF triples with ElasticSearch (explained in chapter 3). One of the initial systems that used IR system to query RDF data was Falcon [41]. In this paper, each document is mapped to with the textual description of the maximum subset of connected RDF triples. In contrast, our work uses URIs for mapping and querying the triples. The ranking of the documents for mapping is based on two factors. The first factor is cosine similarity which is used for mapping keyword terms to documents. The other factor for ranking is the popularity of each document. In our system we used three methods, fuzzy matching, TF-IDF and charNgram for similarity measure, and used the ElasticSearch score function for ranking the answers.

The author of [27] states that several related systems are evaluated in the entity search track of the SemSearch10 workshop [42]. These systems show variations in the TD-IDF weighting adapted for RDF data and returned a ranked list of entities. Although we follow a similar approach in our implementation, these systems are implemented on different datasets. [43] is an

approach that uses inverted lists over terms that appear as entities, the keyword query is translated to a logical expression that returns the URIs of the matching entities; in contrast we indexed used ElasticSearch queries to retrieve the answers. [44] is the work that makes use of ElasticSearch, which is a text-based entry point to the Linked Data cloud. ElasticSearch was also used for indexing and querying Linked Bibliographic Data in JSON-LD format [45]. [45] introduces a solution for representing and indexing bibliographic resources by retaining the integrity and extensibility of Linked Data also supporting fast, customizable indexes in an application-friendly data format. The methodology uses JSON-LD to represent RDF graphs in JSON, which is suitable for indexing with ElasticSearch. Unlike this approach, we applied these methods to convert ontologies and RDF triples to JSON format and we query the data related to a confined domain rather than using a generic dataset.

3. ARCHITECTURE AND IMPLEMENTATION

Figure 2 presents the architecture (AQuA) we used for this research work. In this section, we briefly described each process and specify the details of our implementation.

3.1. Ontology Augmentation

Ontology augmentation process involves selecting, extracting, and reorganizing content from various sources to produce an ontology meeting the specifications of a particular domain and/or task. [46] defines ontology augmentation as the process of enriching an ontology by: i) incorporating new concepts, instances and relations from external resources which include, other ontologies, text, and databases etc., and ii) adding axioms and properties to the ontology. We studied the related work on ontologies related to homelessness and gender factors. The CODO [47] ontology is a COVID-19 ontology which is a data model for publishing COVID-19 data on the web as a knowledge graph. The Homelessness and Clinical Data Recording [48] ontology is a respected conceptual framework used to define degrees of housing insecurity and homelessness internationally. The Gender, Sex, and Sexual Orientation (GSSO) ontology [49] is aimed at bridging gaps between linguistic variations inside and outside the healthcare environment.

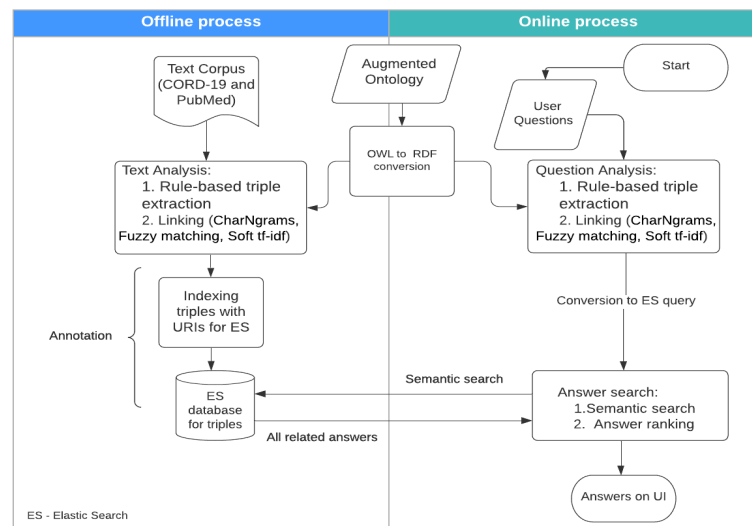


Figure 2: AQuA – Architecture for Question and Answering

These ontologies are used to cover their respective domains but so far, the ontologies are not used to build a QA system or to investigate COVID-19. To produce an ontology meeting the

specifications of this domain to obtain the domain coverage we merged and augmented the above-mentioned ontologies. In general, to meet the specifications of a new task, candidate ontologies may be either manually modified or semi-automatically extended by exploiting different types of information sources. This task is done manually for this research work; there are other ways to accomplish this task which are detailed in [46]. The ontologies are merged using the protégé tool. We also added some new concepts and relations along with equivalency axioms where needed to get a taxonomy which sufficiently covers the domain of two factors selected. 'Reasoning' is an important concept while working with ontologies. The reasoner is a software which is used to understand logical consequences from a collection of asserted facts or axioms. We used a Pellet reasoner while merging and augmenting the ontology. Enabling the reasoner before merging the ontologies ensures that the augmenting task is performed properly. For instance, there are no problems arising from definitions of relations, axioms, and equivalence classes; the super class and subclass hierarchy is correct, and the ontology is consistent.

3.2. Data extraction (text corpus)

Data extraction is the process of extracting domain specific textual data, in our case documents, which will be used for creating triples. To this end, (i) we manually extracted some documents from Kaggle - CORD-19 [50] database which are related to all SDoH factors and filtered out the documents related to homelessness and gender from the collected documents manually and (ii) collected more documents related to homelessness and gender from the PubMed library [51].

There are two main processes involved in our QA architecture (shown in Figure 2): 'Offline Process' and 'Online Process'.

3.3. Offline process

In the offline process, we extract triples from the corpus and store them in a database. The corpus used in this process is natural language text documents. This process consists of two major steps: a) Text analysis on text corpus which includes 2 components namely, a rule-based method for extracting triples from sentences and linking. b) ElasticSearch indexing and storing the RDF triples in an ElasticSearch database

Various processes from the NLP pipeline are applied to the corpus to extract triples. In rule-based triple extraction process, pre-processing is the first step to break the documents to words as machines cannot understand the document as is. For this, we used spaCy, pdfminer and NLTK python libraries which perform operations like parsing/analysing the text, tokenization, POS tagging, etc., over pdf files. After pre-processing, we created patterns to recognise entities and relations between entities. Patterns are created for triple extraction. E.g., for entities: [{'POS': 'NOUN'}; {'POS': 'NOUN'}] which means, any noun followed by a noun can together be considered as an entity; example: corona vaccine. For relations: [{'POS': 'VERB'}] which means any verb is considered as relation/predicate; example: in the sentence 'He ate the mango', ate (verb) is the relation. The advantages of rule-based information extraction technologies are, they are declarative, easy to comprehend, easy to maintain, easy to incorporate with domain knowledge and errors are easy to fix [52]. There are also some disadvantages as this process is heuristic and requires tedious manual labour.

Rule-based Information Extraction (IE) such as triple extraction is valued in the commercial world for its interpretability, which makes IE programs easier to adopt, understand, debug, and maintain in the face of changing requirements [52]. Additionally, rule-based IE is valued as it allows researchers/users to incorporate domain knowledge easily. Other methods to perform NER and RE tasks are based on machine-learning or hybrid methods (combining rule-based and

machine learning techniques). Though there are some pros for these methods, e.g., Machine learning and hybrid methods are trainable, adaptable, and reduce manual efforts, there are also some drawbacks. For instance, these methods require labelled data, the models need to be re-trained for domain adaption, and they are opaque [52]. Therefore, we adopted rule-based IE in our implementation because the domain (Impacts of SDoH on COVID-19) we choose is still evolving (and new social factors affecting Covid-19 are being discovered since the spread of Covid-19) and the data preparation for this task is tedious and requires some more manual support which makes the machine learning and hybrid methods difficult to implement.

There are many python libraries available to perform various NLP tasks. The fundamental purpose of python-based NLP libraries is to simplify text pre-processing. Also, python offers some powerful libraries for leveraging the power of NLP in research projects. We discuss two of the libraries below. spaCy is a python-based library [53] that has many tools that can be applied for various text processing applications in multiple languages. SpaCy uses deep neural networks and transformers in the background. The spaCy library supports various features like Tokenization, POS-Tagging, Dependency Parsing, NER, Rule-based Matching, Text Classification, Linking, etc. [54]. Natural Language ToolKit (NLTK) is another python-library that enables users to build applications in human languages. NLTK is a library for processing string data, which takes a string as input and the output returned is a single or a list of strings. Although spaCy performs better than NLTK in many tasks, NLTK outperforms spaCy in sentence tokenization (as mentioned in [55]). Therefore, we used NLTK for the sentence tokenization task.

Rule-based triple extraction method is done using human created patterns. We used the spaCy library for this operation. We created patterns manually based on scenarios we were interested in (homelessness and gender integrated with transmission and outcomes of COVID-19). We created patterns to extract subjects, predicates, and objects. We extracted 106599 triples from the filtered corpus using these patterns, some are valid, and some are invalid triples. Let us consider the sentence to understand valid triples "Individuals of low incomes are disproportionately likely to suffer from poor mental health". A valid triple extracted for this sentence, and which makes sense is shown in Table 1. Invalid triples are generated as the corpus have different types of values like special characters, numerical values, headers, footers etc. An example demonstrating an invalid triple is shown in Table 2. These invalid triples are filtered while indexing the triples. This filtering task can also be done using data cleaning method which we did not implement at this time.

Table 1: Valid triple

361	low incomes	suffer	poor mental health	Individuals of low incomes are disproportionately likely to suffer from poor mental health.
-----	-------------	--------	--------------------	---

Table 2: Invalid triple

1192	¬©	CMAJ	8(4)	DOI:10.9778/cmajo.20200213 ¬© 2020 Joule Inc. or its licensors CMAJ OPEN 8(4) E627 OPENquantifying heterogeneity in, Áúwhat has happened, Äù a process often referred to as an epidemic appraisal.
------	----	------	------	---

Linking deals with matching the entities (subject/object) and relations (predicates) recognized earlier with the ontology classes (concepts) and relations. Since we are working with two processes (offline and online), interoperability between these processes is crucial; this is improved by the use of ontologies. For this purpose, we need to compare terms (subjects or predicates or objects) from extracted triples to the ontology concepts and relations using

similarity models. In general, concepts and relations in the knowledge base (ontology) have unique identifiers (URIs). The ontologies are converted from OWL file format to RDF format for comparison and mapping using "RDFlib" python library which enables users to perform operations over RDF data. Later we use the same library to convert the RDF file into JSON format so that the ontology concepts and relations along with their respective URIs are indexed and stored in ElasticSearch. This indexing is important for mapping URIs to the terms in triples.

The next step in linking process is to determine the similarity between terms of triples (subjects or predicates or objects) and ontology concepts and relations to map the URIs. To find the similarity between two strings, we built three layers of comparison metrics, CharNgrams, Levenshtein distance/Fuzzy matching, Soft tf-idf. The application finds all the scores and compares them with the defined threshold of 0.9 (to show strong similarity). If any of the three layers produces a match with a score more than the threshold, then the application will map the URI associated with the ontology concepts and relations to the terms of the triples using ElasticSearch. Once the match is found, a keyword-based search is performed using ElasticSearch to fetch the URI and map it to the triple term. This linking process is demonstrated in Figure 3.

ElasticSearch provides an easy way to index documents or triples, enabling users to quickly query the nearest neighbour using a similarity metric. The semantic search (using URIs) is performed over this database to retrieve the answers for the questions in the online process. All the triples which have URIs mapped for subject, and object are filtered out first before indexing. This task is performed to avoid storing lot of data in the database which may hamper the time taken for answer retrieval. After the filtration process, we created URIs for those predicates which do not already have URIs as there was no match with any of the relations in the ontology. To create URIs for the predicate, we used the word as is unlike the regular URIs (Example - The URI created for the word Flourished would be Flourished but not "http://purl.obolibrary.org/obo/Flourished" which is a regular URI pattern). Once we get all the above steps done, all the triples with the mapped URIs are then indexed (using inverted index). The indexed triples are stored as a separate ElasticSearch database. We index and create two databases in our architecture; one is to save ontology concepts and relations with URIs so that this search for mapping URI to the triple term can be facilitated and mapping can be done. The other is to index and store triples, as mentioned above.

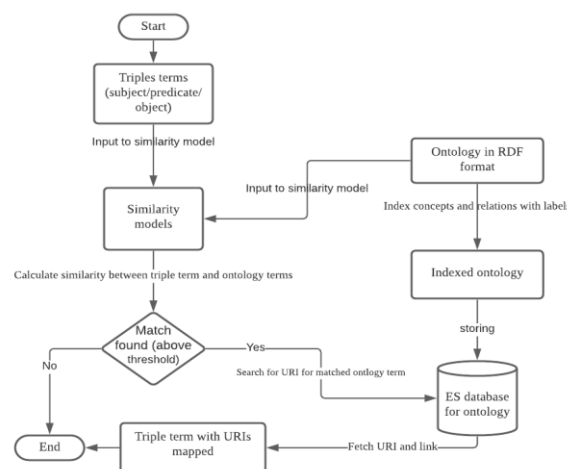


Figure 3: Linking process

3.4. Online process

In online process, we apply question analysis to the natural language question provided to the system in a text format (in natural language) and query the database to retrieve the answer. The online process consists of the following components: a) question analysis, consisting of two NLP methods, namely, a rule-based method of extracting triples from natural language questions and linking the URIs of ontology concepts or relation to terms in the triples; b) converting triples with URIs to ElasticSearch query (question conversion); c) answer search

To generate user questions, we studied various resources available for homelessness and gender [56][57][58][59] and numerous other articles from our dataset to understand the domain better and to create questions to test the application. To this end, we investigated the domain and analysed triples created by the rule-based triple extraction method in offline process and picked some valid triples (20 triples) to create 20 questions on them.

The pre-processing task in the online process is applied on user questions which is the only difference from the offline process. We used the same libraries which are used in the offline process for this task in the online process. The rest of the rule-based triple extraction process is same in both online and offline processes. The same patterns we created in the offline process are used in the online process to extract triples from pre-processed questions. The linking process implementation remains same for both the processes with a small change i.e., we apply the same code in indexing and mapping the URIs with triples extracted from questions from offline process, we do not index the ontology file again in online process.

To perform search over ElasticSearch database, we need ElasticSearch query. We used a boolean query in our implementation. A boolean query is a query that matches answers which match boolean combinations of other queries. The boolean query follows the Lucene BooleanQuery syntax [60]. This query uses one or more boolean clauses. Each clause in the query comes with a typed occurrence. The occurrence types are MUST, SHOULD, MUST NOT, FILTER. The score from each matching term mentioned in the 'must or should clause' will be added to provide the final score for each answer. For this research work, we have used MUST and SHOULD clauses. We create an ElasticSearch query from the mapped triple extracted from the user question by appending subject URI, predicate URI, and object URI. Since we have limited of documents in our data, our questions must be confined to data available. For this reason, we first determined valid triples from the documents and then posed questions. In different situations where web scraping is performed, one would not need to restrict the questions to ensure they contain specific triples.

After generating ElasticSearch queries, we conduct an answer search method involving the NLP methods of semantic search and answer ranking. The word semantic refers to meaning in language or logic. The answer retrieval process is based on semantic search. The annotation and translation processes mentioned above allow a semantic search based on our ontology and term frequency (TF-IDF) weighing schemes. This is to find an answer based on user intent rather than matching the keywords in the search. The factors that guide semantic search are [61]: a) The intent of the user and b) The semantic meaning of search terms. As we have incorporated ontology linking with entities in our project, our search is based on URIs. Usage of URIs guarantees interoperability, which ensures the user's intent is carried over throughout the process. This semantic search is facilitated by using ElasticSearch. The system tries to give alternative answers for the query in our implementation rather than no answer based on user intent. TF-IDF weights also help search engines to have a good idea of what words statistically occur together and make semantic correlations for search and fetch results avoiding spams. The pattern in which this search happens using ElasticSearch query and answer is prioritized is as follows: a) Match all

subject, predicate, and object in question with answer b) Match any two (Subject-predicate, object-predicate, subject-object) in the question with the answer c) Match at least one (subject/predicate/object) in the question with the answer

The final output from our system would be the best-ranked answer from this method. ElasticSearch uses TF-IDF scoring model for ranking the answers, i.e., addition of TF-IDF scores of all the terms in the answers retrieved that matches the question terms (e.g., if answer has 2 terms that match the question term with scores 0.25 and 0.3, the rank would be 0.55). The ranking is done by adding the scores of the less frequent words in the answer retrieved. For example, in the sentence "people suffers from pneumonia", the words people and from are more frequent and the words suffers, and pneumonia are less frequent. Therefore, the subject of the search would be suffers, and pneumonia and the scores of these words are added for ranking. The answers retrieved are sorted in descending order of their ranks.

3.5. User interface (UI)

We used Flask, which provides useful tools and features to create web applications as it is a lightweight python web framework [62][63]. For this purpose, flask uses Jinja templates to dynamically build HTML pages which uses familiar python concepts like variables, loops, lists, etc. These templates are simple files containing static data as well as placeholders for dynamic data. On the UI, the user can enter a question in the search box and click on submit; the system will provide the top 10 related answers on the screen. Figure 4 demonstrates the page layout and design of our QA system.

There is a search bar in the UI through which user can query the system. The query the user searched for appears on the screen with the retrieved answers with search bar for next question. The system provides the metadata of the answers. The field ID provide us the ID of the triple from all the triples extracted in rule-based triple extraction process. The sentence field contain the sentence from which the triple is extracted. The subject, predicate, and object fields provide the information on triples with their mapped URIs. The file field provides the information about the location at which the file is saved; user can go to that location to access the file if needed. The score field gives the rank of the answers retrieved, which are sorted ranks (highest rank first). From the example shown in Figure 4, the rank of the first answer is highest as the URIs we are searching and the URIs retrieved matches. The URIs in the answers from the second position are partially matched with the URIs of the question. As the URIs of the triples from position 3, on, are same and the search and similarity measures use URIs, the score of the answers from position 3, on, are same. The screenshots for all the 20 questions answered are available at https://drive.google.com/drive/folders/1mXaCG-Pmx9m_NWNp-CKUCGxqy4U4X_qm?usp=sharing.

QA System

Query: Why people living in urban informal settlements rely on robust social connections

Extracted Entity URI: https://people.stfx.ca/wmaccaul/Social_Connections / Extracted Predicate URI: <https://people.stfx.ca/wmaccaul/rely>

Sentence	ID	Subject	Predicate	Object	File	Score
Those living in urban informal settlements often rely on robust social connections to survive such as to identify day labor to get food using credit from a street vendor or to find trustworthy child care providers just to name a few [21].	101986	urban informal settlements URI: https://people.stfx.ca/wmaccaul/urban_informal_settlements	rely URI: https://people.stfx.ca/wmaccaul/rely	robust social connections URI: https://people.stfx.ca/wmaccaul/Social_Connections	/Users/priyanka/Desktop/MyComputer/Dataset/Housing//Slum Health- Arresting COVID-19 and Improving Well-Being in Urban Informal Settlements.pdf	35.00148
Those living in urban informal settlements often rely on robust social connections to survive such as to identify day labor to get food using credit from a street vendor or to find trustworthy child care providers just to name a few [21].	101987	often URI:	rely URI: https://people.stfx.ca/wmaccaul/rely	robust social connections URI: https://people.stfx.ca/wmaccaul/Social_Connections	/Users/priyanka/Desktop/MyComputer/Dataset/Housing//Slum Health- Arresting COVID-19 and Improving Well-Being in Urban Informal Settlements.pdf	25.915834
It is known that depression and stress weaken our immune systems.	359	depression URI: https://people.stfx.ca/wmaccaul/depression	weaken URI: https://people.stfx.ca/wmaccaul/weaken	our immune systems URI: https://people.stfx.ca/wmaccaul/immune_system	/Users/priyanka/Desktop/MyComputer/Dataset/Housing//People experiencing homelessness- Their potential exposure to COVID-19.pdf	22.783484
Individuals of low incomes are disproportionately likely to suffer from poor mental health.	361	low incomes URI: https://people.stfx.ca/wmaccaul/Low_Income	suffer URI: https://people.stfx.ca/wmaccaul/suffer	poor mental health URI: https://people.stfx.ca/	/Users/priyanka/Desktop/MyComputer/Dataset/Housing//Slum Health- Arresting COVID-19 and Improving Well-Being in Urban Informal Settlements.pdf	22.783484

Figure 4: User Interface

The domain SDoH is very vast, and the user will have many other questions regarding the impacts of the domain on COVID-19. Our QA system can answer the questions only for the questions from the dataset we have used for this implementation. As we used some portion of the CORD-19 dataset for this implementation, there are some more steps that need to be done for the system to answer a question out of its range (from outside the dataset). They are: a) Gather data with information related to domain and query. b) Extract triples from data using the rule-based triple extraction model (may need more triple patterns). c) Though we have built an ontology related to Homelessness and Gender, users may have questions involving other SDoH factors. For this, the user needs to create ontology classes/relations/instances/axioms according to the requirements. d) Index the triples extracted in step b and store them in Elasticsearch database. The user can then go to the UI and submit the question.

4. EVALUATION

This section describes the evaluation metrics used in measuring the performance of the system, which are the same ones used in IR [64][65]. The implementation is evaluated on the system's ability to correctly retrieve and rank answers for a given question. As our domain for this implementation is novel, there are no benchmark values for these metrics. For a given set of relevant answers also referred to as 'ground truth positive' and a set of answers retrieved by the system also referred to as 'predicted', precision is calculated as per Equations (5).

$$\text{Precision} = \frac{|\{\text{relevant answers}\} \cap \{\text{retrieved answers}\}|}{|\{\text{retrieved answers}\}|} \quad (5)$$

Precision takes all retrieved answers into account [64] giving the fraction of retrieved answers that are relevant. It can also be evaluated considering only the topmost results (Top 5 or top 10 answers for one question) returned by the system using precision@k ($p@k$), where k is the k^{th} ranked answer retrieved by the system for a question q .

Recall is another metric which is often called sensitivity [64]. It is the probability that a relevant answer is retrieved by the query. Recall is calculated as per Equation (6).

$$\text{Recall} = \frac{|\{\text{relevant answers}\} \cap \{\text{retrieved answers}\}|}{|\{\text{relevant answers}\}|} \quad (6)$$

F-measure is a single measure that trades off precision and recall, it is defined as the weighted harmonic mean of precision and recall. This measure gives the accuracy of the model and is calculated as Equation (7).

$$F - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

The above-mentioned measures (precision, recall and F-measure) do not consider the order in which the returned answers are presented, rather they consider only if the relevant answers are retrieved (or not). Whereas the metric precision@k ($p@k$) is used to determine relevancy of the system by considering the topmost answers retrieved.

The Average Precision (AP) is a metric that gives a good measurement of quality, i.e., the ability of the model to appropriately sort the results of a query. AP is calculated as Equation (8).

$$\text{AP} = \frac{1}{\text{GTP}} \sum_{j=1}^N p@j \times \text{rel}@j \quad (8)$$

where N is the total number of positions i.e., the total number of results returned.

For a query, $\text{AP}@k$, the average precision till the k^{th} position in the ranked list, is calculated as in Equation (9).

$$\text{AP}@k = \frac{1}{\text{GTP}} \sum_{j=1}^k p@j \times \text{rel}@j \quad (9)$$

where, $\text{rel}@k$ is the indicator function which equals to 1 if answer at rank k is relevant, 0 otherwise and GTP is the number of ground truth positives. The result of this metric will be '1' if the answers are sorted correctly and all the relevant answers appear at the top of the ranked list. Remark: in general, $\text{AP}@N = \text{AP}$ where N is the last position.

The metrics explained above (Precision, Recall, F-measure, and Average Precision), are based on single question. The metrics: Mean Average Precision, Mean Average Precision@k and system Precision@k, involve all the queries.

The Mean Average Precision, mAP, for a set of queries $\{q_i \mid 1 \leq i \leq Q\}$ is the mean of the AP over all queries q_i ; it is given in Equation (10).

$$mAP = \frac{1}{Q} \sum_{i=1}^Q AP(q_i) \quad (10)$$

The $mAP@k$, mean average precision at the k^{th} position over all queries q_i , is calculated as in Equation (11).

$$mAP@k = \frac{1}{Q} \sum_{i=1}^Q AP@k(q_i) \quad (11)$$

We use $mAP@k$ to evaluate our project as it is the most popular metric to evaluate the performance of any IR systems. $mAP@k$ is a metric which gives a single-figure measure of quality across all queries at the k^{th} position [66].

The system precision@k, $sp@k$, is the averaged precision over all the answers retrieved of all queries till the k^{th} position. This metric is calculated as Equation (12).

$$sp@k = \frac{\sum_{i=1}^Q \sum_{j=1}^k p@j(q_i)}{|\{\text{total retrieved answers}\}|} \quad (12)$$

This metric gives the overall performance of the system till the k^{th} position.

4.1. Results

We calculated $p@k$ for all the positions; Figure 5 shows how the ranking quality of the application is gradually decreasing as the rank of the answer is increasing. Therefore, we decided to show the top 10 answers on the UI for any question as the graph is gradually falling off after position 10.

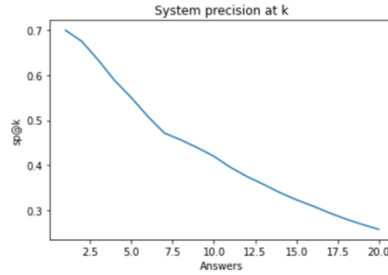


Figure 5: System Precision@k

We used the system precision calculated above along with the recall (which is manually calculated) to obtain F-measure, which gives the accuracy of overall system for top 5 answers retrieved for 20 questions, top 10 answers retrieved for 20 questions, top 15 answers retrieved for 20 questions and top 20 answers retrieved for 20 questions. The Table 3 shows the evaluated results. From the obtained output of F-measure, it is clear that the system's accuracy is increasing till the top 10 answers and falling off from answers 11-20.

Table 3: System Precision, Recall and F-measure

Top n answers	System precision	Recall	F-measure
5	0.55	0.495	0.5298
10	0.419	0.773	0.56644
15	0.3233	0.895	0.495937
20	0.2574	0.947	0.42261526

We evaluated the quality of the system using the mAP@k metric. We conducted these experiments for mAP at each kth position using 'Information Retrieval (IR) Effectiveness Evaluation Library for Python' [68]. This library takes two files as input; one is ground truth (relevant) file, and another is predicted (retrieved) file. We manually analysed the answers and determined which were correct for predicted file and then labeled the data with relevant answer and non-relevant answer to get the ground truth file (for the questions under consideration). The results obtained for mAP@k are shown the graph in Figures 6.

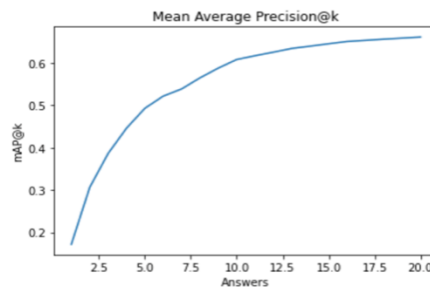


Figure 6: Mean Average Precision@k

Figure 6 shows that mAP@k is increasing indicating the quality of the system is increasing for each question as the number of relevant answers are being retrieved by the system are increasing. Figure 6 shows that mAP@10 i.e., the performance of the system for 20 questions and top 10 answers each is 0.60.

5. CONCLUSIONS AND FUTURE WORKS

In this research, we developed a prototype QA system so that people can get information from a large corpus for questions involving SDoH without reading all the documents. The QA system developed is the integration of semantic web technologies and IR systems. Innovative aspects for this work include using ElasticSearch, which is the most widely used IR systems, instead of SPARQL for querying the database, and creating a domain specific ontology which supports this QA system. To meet the needs of an interdisciplinary audience, we present an overview of various methods involved in the implementation of QA systems such as: how to collect or create domain-specific ontologies, how to collect the domain-specific data, how to create RDF triples using a rule-based approach, how to perform search over RDF triples using ElasticSearch, how to index an RDF dataset, how to rank the data, what data should be ranked and how to evaluate an IR-based QA system using some popular metrics like Mean Average Precision and System Precision. Our system performance can be improved if the future works discussed below are integrated with the existing system. Also, this QA system can be easily adopted to any domain by creating domain specific patterns, collecting domain specific documents and domain specific knowledge graphs or ontologies.

From our literature review on the SDoH domain, we determined that social determinants of health have a major impact on a crisis like COVID. Our goal through this application is to show that we can leverage the existing research and literature on COVID to help decision-makers manage health emergencies so that everyone has an equal opportunity to stay healthy. In future, the ontology we created can be enhanced by adding other factors (other than homelessness and gender) to cover the entire domain of SDoH. We will further investigate how to improve the overall performance of the system by implementing more semantic similarity metrics like cosine similarity for linking and query expansion. Query expansion using ontology parent-child and sibling relations will improve results. An appropriate axiom schema to further enhance the semantics of our search must be developed and integrated into our system. This requires interactions with clinicians, epidemiologists, and other public health professionals to further leverage queries on the existing research and literature on COVID-19. Data relevant to queries to cover all SDoH factors is also needed. Other future work includes data cleaning applied to triple extraction process and getting this application evaluated by clinicians.

REFERENCES

- [1] Dimitrakis, E., Sgontzos, K. and Tzitzikas, Y., 2020. A survey on question answering systems over linked data and documents. *Journal of Intelligent Information Systems*, 55(2), pp.233-259.
- [2] Wang, W., Auer, J., Parasuraman, R., Zubarev, I., Brandyberry, D. and Harper, M., 2000. A question answering system developed as a project in a natural language processing course. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.
- [3] Kolomiyets, O. and Moens, M.F., 2011. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24), pp.5412-5434.
- [4] Athenikos, S.J. and Han, H., 2010. Biomedical question answering: A survey. *Computer methods and programs in biomedicine*, 99(1), pp.1-24.
- [5] Gupta, P. and Gupta, V., 2012. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4).
- [6] Kalyanpur, A., Boguraev, B.K., Patwardhan, S., Murdock, J.W., Lally, A., Welty, C., Prager, J.M., Coppola, B., Fokoue-Nkoutche, A., Zhang, L. and Pan, Y., 2012. Structured data and inference in DeepQA. *IBM Journal of Research and Development*, 56(3.4), pp.10-1.
- [7] Soares, M.A.C. and Parreiras, F.S., 2020. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, 32(6), pp.635-646.
- [8] Chan, H.Y. and Tsai, M.H., 2019. Question-answering dialogue system for emergency operations. *International Journal of Disaster Risk Reduction*, 41, p.101313.
- [9] Covid-19 - what we know so far about...social determinants of health, <https://www.publichealthontario.ca/-/media/documents/ncov/covid-wwksf/2020/05/what-we-know-social-determinants-health.pdf?la=en>, [Online; accessed 19-June-2021].
- [10] Five big lessons experts say canada should learn from covid-19, <https://www.ctvnews.ca/health/coronavirus/five-big-lessons-experts-say-canada-should-learn-from-covid-19-1.5282125>, [Online; accessed 19-June-2021].
- [11] Singu, S., Acharya, A., Challagundla, K. and Byraredddy, S.N., 2020. Impact of social determinants of health on the emerging COVID-19 pandemic in the United States. *Frontiers in public health*, 8, p.406.
- [12] Abacha, A.B. and Zweigenbaum, P., 2015. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies. *Information processing & management*, 51(5), pp.570-594.
- [13] Introduction to natural language processing (nlp), <https://www.kdnuggets.com/2019/10/introduction-natural-language-processing.html>, [Online; accessed 23-May-2021].
- [14] Wikipedia contributors. Question answering. Wikipedia, The Free Encyclopedia. August 14, 2021, 06:46 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Question_answering&oldid=1038707361. Accessed August 26, 2021.

- [15] Prager, J., 2021. Question answering. In *The Oxford Handbook of Computational Linguistics* 2nd edition.
- [16] Srinivasa-Desikan, B., 2018. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.
- [17] Wikipedia contributors. Hidden Markov model. Wikipedia, The Free Encyclopedia. August 5, 2021, 04:02 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Hidden_Markov_model&oldid=1037203864. Accessed August 26, 2021.
- [18] Different ways of doing relation extraction from text, <https://medium.com/@andreasherman/different-ways-of-doing-relation-extraction-from-text-7362b4c3169e>, [Online; accessed 03-July-2021].
- [19] Wikipedia contributors. Ontology. Wikipedia, The Free Encyclopedia. August 21, 2021, 13:25 UTC. Available at: <https://en.wikipedia.org/w/index.php?title=Ontology&oldid=1039903261>. Accessed August 26, 2021.
- [20] Rubin, D.L., Noy, N.F. and Musen, M.A., 2007. Protege: a tool for managing and using terminology in radiology applications. *Journal of digital imaging*, 20(1), pp.34-46.
- [21] Introduction to ontology, <https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/#:~:text=An%20ontology%20is%20a%20formal,relationships%20that%20hold%20between%20them.&text=As%20a%20result%2C%20ontologies%20do,new%20knowledge%20about%20the%20domain>, [Online; accessed 18-June-2021].
- [22] What is an rdf triplestore, <https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf-triplestore/>, [Online; accessed 10-July-2021].
- [23] What's a uri and why does it matter?, <http://www.ltg.ed.ac.uk/~ht/WhatAreURIs/#:~:text=URI%20stands%20for%20Uniform%20Resource,the%20World%20Wide%20Web%20consortium.>, [Online; accessed 10-July-2021].
- [24] Rdf 1.1 primer, <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>, [Online; accessed 10-July-2021].
- [25] Introduction to semantic web, <https://graphdb.ontotext.com/documentation/standard/introduction-to-semantic-web.html>, [Online; accessed 18-June-2021].
- [26] Olsson, J., 2019. Using Elasticsearch for full-text searches on unstructured data.
- [27] Kadilierakis, G., Fafalios, P., Papadakos, P. and Tzitzikas, Y., 2020, May. Keyword search over RDF using document-centric information retrieval systems. In *European Semantic Web Conference* (pp. 121-137). Springer, Cham.
- [28] Elasticsearch as an ir tool, https://colab.research.google.com/github/fastforwardlabs/ff14_blog/blob/master/_notebooks/2020-06-30-Evaluating_the_Retriever_&_End_to_End_System.ipynb#scrollTo=v1MjzT9zlTrf, [Online; accessed 10-June-2021].
- [29] Berglund, P., 2014. Shard Selection in Distributed Collaborative Search Engines A design, implementation and evaluation of shard selection in Elasticsearch.
- [30] Wu, G., He, Y. and Hu, X., 2018. Entity linking: an issue to extract corresponding entity with knowledge base. *IEEE Access*, 6, pp.6220-6231.
- [31] The levenshtein-algorithm, <http://www.levenshtein.net/>, [Online; accessed 12-June-2021].
- [32] Article on charngram - pytorch, https://pytorchnlp.readthedocs.io/en/latest/_modules/torchnlp/word_to_vector/char_n_gram.html, [Online; accessed 16-June-2021].
- [33] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, pp.135-146.
- [34] Green Jr, B.F., Wolf, A.K., Chomsky, C. and Laughery, K., 1961, May. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference* (pp. 219-224).
- [35] Woods, W.A. and WA, W., 1977. Lunar rocks in natural English: Explorations in natural language question answering.
- [36] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J. and Schlaefel, N., 2010. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3), pp.59-79.
- [37] Song, S., Zhang, X. and Qin, G., 2017. Multi-domain ontology mapping based on semantics. *Cluster Computing*, 20(4), pp.3379-3391.

- [38] Anam, S., Kim, Y.S., Kang, B.H. and Liu, Q., 2016, February. Adapting a knowledge-based schema matching system for ontology mapping. In *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1-10).
- [39] Pham, M., Alse, S., Knoblock, C.A. and Szekely, P., 2016, October. Semantic labeling: a domain-independent approach. In *International Semantic Web Conference* (pp. 446-462). Springer, Cham.
- [40] Ramnandan, S.K., Mittal, A., Knoblock, C.A. and Szekely, P., 2015, May. Assigning semantic labels to data sources. In *European Semantic Web Conference* (pp. 403-417). Springer, Cham.
- [41] Cheng, G. and Qu, Y., 2009. Searching linked objects with falcons: Approach, implementation and evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), pp.49-70.
- [42] Liu, X. and Fang, H., 2010, April. A study of entity search in semantic search workshop. In *Proc. of the 3rd Intl. Semantic Search Workshop*.
- [43] Delbru, R., Campinas, S. and Tummarello, G., 2012. Searching web data: An entity retrieval and high-performance indexing model. *Journal of Web Semantics*, 10, pp.33-58.
- [44] Ilievski, F., Beek, W., van Erp, M., Rietveld, L. and Schlobach, S., 2016, May. LOTUS: Adaptive text search for big linked data. In *European Semantic Web Conference* (pp. 470-485). Springer, Cham.
- [45] Johnson, T., 2013. Indexing linked bibliographic data with JSON-LD, BibJSON and Elasticsearch. *Code4lib Journal*, (19).
- [46] Fernandez, M., Zhang, Z., Lopez, V., Uren, V. and Motta, E., 2011, June. Ontology augmentation: combining semantic web and text resources. In *Proceedings of the sixth international conference on Knowledge capture* (pp. 9-16).
- [47] Dutta, B. and DeBellis, M., 2020. CODO: an ontology for collection and analysis of COVID-19 data. *arXiv preprint arXiv:2009.01210*.
- [48] Homelessness and clinical data recording, <https://bioportal.bioontology.org/ontologies/HCDR/?p=summary>, [Online; accessed 19-June-2021].
- [49] Gsso, <https://bioportal.bioontology.org/ontologies/GSSO>, [Online; accessed 19-June-2021].
- [50] Wang, L.L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R., Liu, Z., Merrill, W. and Mooney, P., 2020. Cord-19: The covid-19 open research dataset. *ArXiv*.
- [51] Pubmed library - national library of medicine, <https://pubmed.ncbi.nlm.nih.gov/?term=covid-19+homeless+canada&filter=simsearch3.fff>, [Online; accessed 21-June-2021].
- [52] Chiticariu, L., Li, Y. and Reiss, F., 2013, October. Rule-based information extraction is dead! long live rule-based information extraction systems!. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 827-832).
- [53] Honnibal, M. and Montani, I., 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1), pp.411-420.
- [54] spacy 101: Everything you need to know, <https://spacy.io/usage/spacy-101>, [Online; accessed 16-June-2021].
- [55] Introduction to libraries of nlp in python - nltk vs. spacy, <https://medium.com/@akankshamalhotra24/introduction-to-libraries-of-nlp-in-python-nltk-vs-spacy-42d7b2f128f2>, [Online; accessed 16-June-2021].
- [56] Social determinants of health, <https://www.healthypeople.gov/2020/topics-objectives/topic/social-determinants-of-health>, [Online; accessed 30-April-2021].
- [57] Braveman, P. and Gottlieb, L., 2014. The social determinants of health: it's time to consider the causes of the causes. *Public health reports*, 129(1_suppl2), pp.19-31.
- [58] Nchhstp social determinants of health, <https://www.cdc.gov/nchhstp/socialdeterminants/faq.html>, [Online; accessed 30-April-2021].
- [59] Artiga, S., Orgera, K. and Pham, O., 2020. Disparities in health and health care: Five key questions and answers. Kaiser Family Foundation.
- [60] Boolean query, <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-bool-query.html>, [Online; accessed 18-June-2021].
- [61] Semantic search, <https://blog.alexa.com/semantic-search/>, [Online; accessed 18-June-2021].
- [62] Building modern user interfaces with ask, <https://ckraczkowsky.medium.com/building-modern-user-interfaces-with-flask-23016d453792>, [Online; accessed 22-June-2021].
- [63] How to make a web application using ask in python3, <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>, [Online; accessed 22-June-2021].

- [64] Wikipedia contributors. Evaluation measures (information retrieval). Wikipedia, The Free Encyclopedia. July 30, 2021, 08:39 UTC. Availableat: [https://en.wikipedia.org/w/index.php?title=Evaluation_measures_\(information_retrieval\)&oldid=1036232908](https://en.wikipedia.org/w/index.php?title=Evaluation_measures_(information_retrieval)&oldid=1036232908). Accessed August 26, 2021.
- [65] Breaking down mean average precision (map), <https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52>, [Online; accessed 21-June-2021].
- [66] Evaluation in information retrieval, <https://nlp.stanford.edu/IR-book/pdf/08eval.pdf>, [Online; accessed 21-June-2021].
- [67] Improving entity resolution with soft tf-idf algorithm, <https://medium.com/enigma-engineering/improving-entity-resolution-with-soft-tf-idf-algorithm-42e323565e60>, [Online; accessed 16-June-2021].
- [68] Information retrieval (ir) effectiveness evaluation library for python, <https://pypi.org/project/ir-evaluation-py/>, [Online; accessed 22-June-2021]

AUTHORS

Priyanka Addagudi is a Graduate Research Assistant at St Francis Xavier University, Canada. Her interest areas and research work include building information retrieval-based QA system which involves semantic search engines, knowledge graphs, NLP, modern deep learning architectures, and statistical language modeling.

Wendy MacCaull is a Senior Research Professor in Computer Science at St Francis Xavier University, Canada. Her research interests include knowledge representation and reasoning, ontologies, non-classical logics, and workflow systems with applications to healthcare.

ESGBERT: LANGUAGE MODEL TO HELP WITH CLASSIFICATION TASKS RELATED TO COMPANIES' ENVIRONMENTAL, SOCIAL, AND GOVERNANCE PRACTICES

Srishti Mehra*, Robert Louka*, Yixun Zhang

University of California Berkeley, School of Information, USA

ABSTRACT

Environmental, Social, and Governance (ESG) are non-financial factors that are garnering attention from investors as they increasingly look to apply these as part of their analysis to identify material risks and growth opportunities. Some of this attention is also driven by clients who, now more aware than ever, are demanding for their money to be managed and invested responsibly. As the interest in ESG grows, so does the need for investors to have access to consumable ESG information. Since most of it is in text form in reports, disclosures, press releases, and 10-Q filings, we see a need for sophisticated natural language processing (NLP) techniques for classification tasks for ESG text. We hypothesize that an ESG domain specific pre-trained model will help with such and study building of the same in this paper. We explored doing this by fine-tuning BERT's pre-trained weights using ESG specific text and then further fine-tuning the model for a classification task. We were able to achieve accuracy better than the original BERT and baseline models in environment-specific classification tasks.

KEYWORDS

ESG, NLP, BERT, Universal Sentence Encoder, Deep Averaging Network

1. INTRODUCTION

The importance of Environmental, Social, and Governance (ESG) issues has risen in prominence over the last decade. In the early 1990's, fewer than 20 publicly listed companies issued reports that included ESG data; that number grew to almost six thousand by 2014 [1]. Regulations for SEC filings to follow certain standards for responsibility about Climate Change and Human Governance, and Investor and Shareholder support has driven the motivation for these disclosures.

There has been little research analyzing the non-financial information content [2] in financial disclosures. The most common methods for analyzing the non-financial, narrative information content remain manual or dictionary-based [3][4][5][6][7][8]. The existing literature focuses on the quantity of non-financial information published, rather than its content [9][10]. This underpins the requirement for a study like ours.

Domain-specific BERT variations like FinBERT [11] and BioBERT [12], that have been either fine-tuned or pre-trained on domain corpus instead of or in addition to the generic English language, have achieved great results in studying the information content, particularly for domain specific language tasks. The primary interest of this research is to harness that benefit for ESG

specific text classification tasks. We study building an environment-specific variation of BERT by fine-tuning the pre-trained BERT weights using a Masked Language Model (MLM) task on an ESG corpus and then further fine-tuning our model for Sequence Classification to predict:

1. A change or no change in environmental scores, and
2. A positive or negative change (if any) in environmental scores of companies using ESG related text in their 10-Q filings.

2. BACKGROUND

Accounting for Sustainability (<https://www.accountingforsustainability.org/>) is a project that aims to inspire action by finance leaders to drive a fundamental shift towards resilient business models and a sustainable economy. To do so the project publishes guides, case studies, blogs, reports and surveys, and hosts webinars. This material is available on their knowledge hub and is reflective of the opportunities and risks posed by environmental and social issues. These are what we chose as our ESG corpus to pre-train our BERT model on top of the English Wikipedia and BooksCorpus it has been trained on [13].

In 2010, the SEC published an interpretive release on climate change-related disclosures “to remind companies of their obligations under existing federal securities laws and regulations to consider climate change and its consequences as they prepare disclosure documents” [14]. Therefore, the company’s disclosures should not only consist of financial narratives but also contain information about the environmental aspects concerning the firm [9]. This is the reason we chose to use 10-Q filings as our input for the classification task.

Sustainalytics’ ESG Risk Ratings measure a company’s exposure to industry-specific material ESG risks and how well a company is managing those risks. This multi-dimensional way of measuring ESG risk combines the concepts of management and exposure to arrive at an absolute assessment of ESG risk. These risk scores are also broken down into environmental, social, and governance risks. Of these, for our research, we use the change in total environmental risk score for each company quarter over quarter to indicate whether there was: 1. A change or no change, and 2. A positive or negative change.

3. RELATED WORK

This section describes previous research conducted on domain-specific variations of BERT (3.1) and ESG related NLP research (3.2).

3.1. Domain-specific BERT variants

FinBERT [11] author explored pre-training BERT on Financial corpus based on their learning from a previous study by Howard and Ruder [15] which shows that further pre-training a language model on a target domain corpus improves the eventual classification performance. They pretrained BERT on finance-specific corpora and used those weights to further train the model for financial sentiment classification. They saw improved results in comparison to the original BERT (pre-trained on generic English language corpora).

BioBERT [12] authors, similarly (similar architecture as followed by FinBERT), pre-trained BERT with Biomedical corpora in addition to the English language corpora it was already trained on. They went on to find that BioBERT largely outperformed BERT in a variety of biomedical text mining tasks.

3.2. ESG related NLP research

Armbhurst, Schäfer, and Klinger, 2020 studied the effect of the environmental performance of a company (as learned from MD&A sections in 10-K and 10-Q filings) on the relationship between the company's disclosures and financial performance. They found that textual information contained within the MD&A section does not allow for conclusions about the future (corporate) financial performance. However, there is evidence that the environmental performance can be extracted by NLP methods.

Serafeim and Yoon [1] showed that ESG ratings predict future ESG news and market reactions, particularly when there is disagreement amongst raters. This study is similar to our study in that it uses ESG scores (from TruValue, a company similar to Sustainalytics, which we use for ESG scores) to predict the public reaction, whereas we are using information from public documents to predict ESG scores. The news in their study was aggregated by TruValue using machine learning enabled text mining from a wide variety of sources.

4. METHOD

This section will be divided into BERT (4.1), pre-trained BERT weights on ESG specific corpus (4.2), and fine-tuning for the classification task mentioned in earlier sections (4.3).

4.1. Bidirectional Encoder Representations from Transformers (BERT)

BERT [13] is a pre-trained model that builds word representations learned through bi-directional tasks. They use a Masked Language Model (MLM) task to fuse the left and the right context, which allows them to pretrain a deep bidirectional Transformer. The task randomly masks some of the tokens from the input and predicts the masked words based only on context. Additionally, they use the Next Sentence Prediction (NSP) task that captures the relationship between two sentences which is not directly captured by language modeling. For finetuning, the BERT model is first initialized with the pre-trained parameters learned in the bi-directional approach, and those parameters are then fine-tuned using labeled data from the downstream tasks. Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters [13].

4.2. Pre-training on ESG Specific Corpus

BERT's pre-training procedure largely follows the existing literature on language model pretraining, they use the BooksCorpus (800M words) [16] and English Wikipedia (2,500M words) for the same.

Since the text in our research is also in the English language, we did not want to forgo the benefit of pre-trained weights on such large English language corpora. Thus, we further train BERT's pre-trained weights using an additional Masked Language Modeling (MLM) Task.

We use text from the Knowledge Hub of Accounting for Sustainability for our MLM task. These occur in the form of guides, case studies, blogs, reports, and surveys. We tokenized the text found in those documents using BERT's WordPiece Tokenizer, masked 15% of the words, and learned to predict those masked words. In doing so, we updated the pre-trained weights of BERT to reflect learnings from ESG context. We chose the MLM task since it learns to predict the masked words based only on context.

4.3. Fine-Tuning for Classification Task

In order to test our hypothesis of a domain-specific variation of BERT working better than that original trained on generic language, we chose two classification tasks that we fine-tuned on. The classification tasks were to predict whether there was:

1. A change or no change, and
2. A positive or negative change (if any) in environmental scores of companies using ESG related text in their 10-Q filings.

4.3.1. Input for Fine-Tuning with Classification

BERT takes up to 512 tokens as input. Since our input per company per quarter was an entire 10Q document, which are multiple pages long, we needed a way to choose 512 tokens from each document. 10-Q reports contain small portions that address environmental factors. Therefore, our approach was to extract the sentences most relevant to environmental factors and choose 512 tokens from those. In order to do so, we needed a method to order all sentences in the report (or pick the top 3) by relevance. We encoded sentences in the reports and compared them using cosine similarity with a few benchmark sentences that we thought would help us extract the most relevant sentences from these documents.

For the encoding of the sentences, we experimented with Sentence BERT [17] and Universal Sentence Encoder [18]. We found that the Deep Averaging Network (DAN) version of the Universal Sentence Encoder works in this case, to extract the most relevant sentences. Since we were using the DAN version of the Universal Sentence Encoder, we created our benchmark sentence as one that was a scramble of words that reflect high relevance with environment factors. We hypothesize that the scramble of words helped us extract a deeper, more diverse set of relevant sentences from the documents.

After encoding and comparing each sentence in the report with the benchmark sentence(s), we chose the top 3 sentences for each document and fed that as input to our model. We let there be a truncation for those that exceeded 512 tokens and padding for those that had less than 512 tokens in the 3 sentences chosen.

4.3.2. Fine-Tuning

The approach for fine-tuning for both the classification tasks was to use BERT embeddings and attention masks of the chosen 512 tokens and feed them into the model that was fine-tuned on the ESG corpus. The outputs of that were used as inputs for a classification layer to learn ESG scores. The outputs for the classification layer were scores from Sustainalytics for each company for each. The models were hyperparameter tuned to achieve the results discussed in section 6.

5. DATA

5.1. ESG Corpus for Fine-Tuning

The ESG corpus that we fine-tuned BERT's pre-trained weights on was obtained from the Knowledge Hub of the Accounting for Sustainability project.

5.2. Input and Output for Classification

For our input, we got 10-Q reports for S&P 500 companies from University of Notre Dame's Software Repository for Accounting and Finance for the time frame 2014-2018. For the output, we used Wharton's research platform WRDS to obtain quarterly Sustainability scores for the same companies for the same time frame.

5.3. EDA

The distribution of scores (Figure 1) does not vary highly. Roughly 60% of the quarterly changes in environmental scores are zero. While the tails of the distribution do contain score changes on the larger side, most of the changes are quite small. This does not affect our architecture much since we are doing binary classifications (change or no change; positive change or negative change). Figure 2 shows the relative frequency of the sentence lengths which was used to decide our truncation and padding strategy to ensure we feed our model 512 tokens each time.

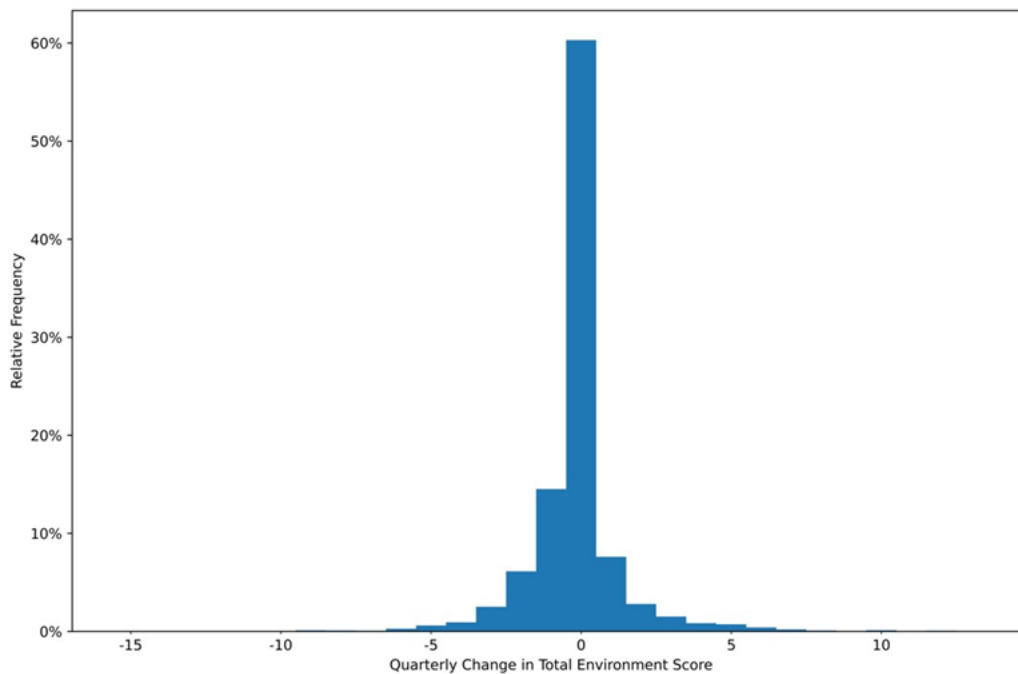


Figure 1. Quarterly change in Total Environment Scores for companies

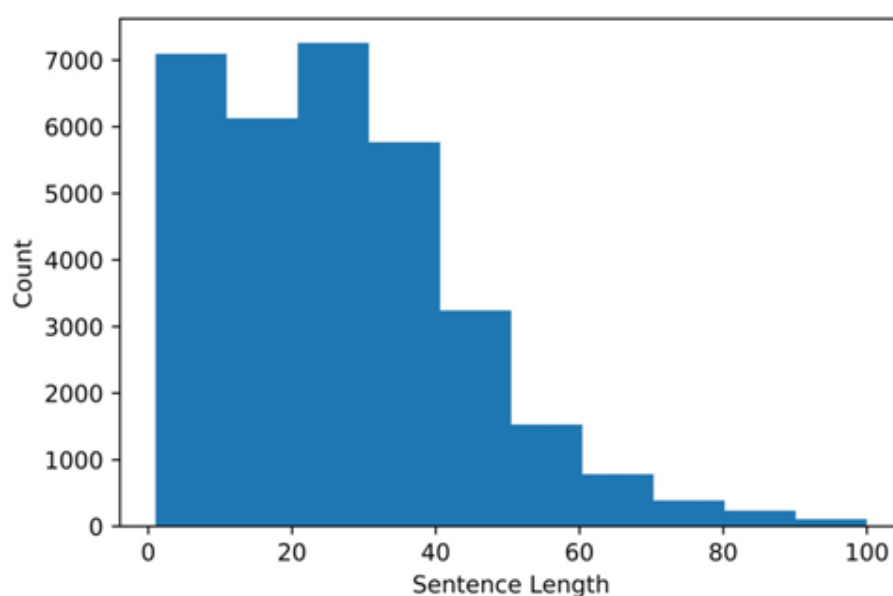


Figure 2. Histogram of length of sentences in the 10-Q filings

6. RESULTS

Naive Bayes barely beat the common class in training and could not beat it in the validation or test data sets. ESGBERT is able to outperform BERT and the other classification techniques we compared against in both tasks that we trained our model on. The results are illustrated in Table 1 and Table 2 respectively, and details about the relevant hyperparameters are mentioned in the description below each table.

Table 1. Classification for change or no change in environmental risk score of company per quarter. Models run with learning rate 2e-05, epsilon 1e-08, 8 epochs, and batch size of 8.

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Common Class Prediction	0.6107	0.614	0.5791
BERT	0.6251	0.6325	0.5985
ESGBERT	0.839	0.7906	0.6709

Table 2. Classification for positive or negative change in environmental risk score of company per quarter. Models run with learning rate 2e-05, epsilon 1e-08, 8 epochs, and batch size of 8.

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Common Class Prediction	0.5974	0.5978	0.5682
BERT	0.6583	0.6055	0.4317
ESGBERT	0.8618	0.8	0.793

7. CONCLUSION

With our research, we strengthen confidence in the learning that pre-training BERT on domain specific corpus yields better results in classification tasks related to that domain. We anticipate that ESGBERT's pre-trained weights, that have learned ESG context, can be used for multiple ESG specific text classification tasks and researchers/developers will benefit from them. For example, our model's pre-trained weights can be used to predict Social and Governance risk scores for companies in addition to the Environmental risk scores that we predicted.

Additionally, the weights can be enhanced by training on additional ESG corpora, like ESG disclosures that companies have now started to release. Since such disclosures will fully focus on the Environmental, Social, and Governance practices and investments, they will have more to inform about the scores than 10-Q filings did.

REFERENCES

- [1] Serafeim, George and Yoon, Aaron, Stock Price Reactions to ESG News: The Role of ESG Ratings and Disagreement (January 13, 2021). Harvard Business School Accounting & Management Unit Working Paper No. 21-079, Available at SSRN: <https://ssrn.com/abstract=3765217> or <http://dx.doi.org/10.2139/ssrn.3765217>
- [2] Kölbel, J., Leippold, M., Rillaerts, J., & Wang, Q. (2020). Does the CDS market reflect regulatory climate risk disclosures?. SSRN, (3616324)
- [3] Berkman, H., Jona, J., & Soderstrom, N. S. (2019). Firm-specific climate risk and market valuation. Available at SSRN 2775552
- [4] Matsumura, E. M., Prakash, R., & Vera-Muñoz, S. C. (2018). Capital market expectations of risk materiality and the credibility of managers' risk disclosure decisions. Available at SSRN, 2983977
- [5] Reverte, C. (2016). Corporate social responsibility disclosure and market valuation: evidence from Spanish listed firms. *Review of Managerial Science*, 10(2), 411-435
- [6] Verbeeten, F. H., Gamerschlag, R., & Möller, K. (2016). Are CSR disclosures relevant for investors? Empirical evidence from Germany. *Management Decision*
- [7] Clarkson, P. M., Li, Y., Richardson, G. D., & Vasvari, F. P. (2008). Revisiting the relation between environmental performance and environmental disclosure: An empirical analysis. *Accounting, organizations and society*, 33(4-5), 303-327
- [8] Cormier, D., & Magnan, M. (2007). The revisited contribution of environmental reporting to investors' valuation of a firm's earnings: An international perspective. *Ecological economics*, 62(3-4), 613-626
- [9] Armbrust, F., Schäfer, H., & Klinger, R. (2020, December). A Computational Analysis of Financial and Environmental Narratives within Financial Reports and its Value for Investors. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation* (pp. 181-194)
- [10] Hummel, K., & Schlick, C. (2016). The relationship between sustainability performance and sustainability disclosure—Reconciling voluntary disclosure theory and legitimacy theory. *Journal of accounting and public policy*, 35(5), 455-476
- [11] Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063
- [12] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pretrained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234-1240
- [13] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
- [14] SEC. 2010. Commission guidance regarding disclosure related to climate change. <http://www.sec.gov/rules/interp/2010/33-9106.pdf>
- [15] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146

- [16] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision (pp. 1927)
- [17] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bertnetworks. arXiv preprint arXiv:1908.10084
- [18] Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Kurzweil, R. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175

AUTHOR

Srishti Mehra, Robert Louka, and Yixun Zhang are graduate students of University of California, Berkeley, studying/studied Masters in Information and Data Science.



DEEP LEARNING FRAMEWORK MINDSPORE AND PYTORCH COMPARISON

Xiangyu XIA and Shaoxiang ZHOU

Department of Information, Beijing City University, Beijing 100191, China

ABSTRACT

Deep learning has been well used in many fields. However, there is a large amount of data when training neural networks, which makes many deep learning frameworks appear to serve deep learning practitioners, providing services that are more convenient to use and perform better. MindSpore and PyTorch are both deep learning frameworks. MindSpore is owned by HUAWEI, while PyTorch is owned by Facebook. Some people think that HUAWEI's MindSpore has better performance than FaceBook's PyTorch, which makes deep learning practitioners confused about the choice between the two. In this paper, we perform analytical and experimental analysis to reveal the comparison of training speed of MindSpore and PyTorch on a single GPU. To ensure that our survey is as comprehensive as possible, we carefully selected neural networks in 2 main domains, which cover computer vision and natural language processing (NLP). The contribution of this work is twofold. First, we conduct detailed benchmarking experiments on MindSpore and PyTorch to analyze the reasons for their performance differences. This work provides guidance for end users to choose between these two frameworks.

KEYWORDS

Deep Learning, Performance, Mindspore, Pytorch, Comparison.

1. INTRODUCTION

In recent years, machine learning has been used in various fields. Deep learning is a machine learning method that has many applications in computer vision, speech recognition, natural language processing and other fields. Deep learning relies on the number of neurons and layers of the neural network. With the continuous improvement of the depth of the network and the number of neurons, the relative accuracy will be higher. From traditional fully connected neural networks to convolutional neural networks and recurrent recurrent neural networks to graph neural networks and transformers, the depth and complexity of networks are getting higher and higher, and the types of networks are constantly changing. The more complex the network, the more parameters for training, which requires huge computing power, which requires a large number of high-performance computing cards to accelerate, such as GPU, TPU [1] and FPGA. Among these hardware options, GPU is the most popular choice. As neural networks become more and more complex, the continuous upgrade and iteration of computing cards has prompted developers of deep learning frameworks to provide users of deep learning with better usability and higher performance. Deep Learning Framework.

There are currently many deep learning frameworks, including Caffe [2] developed by UC Berkeley, TensorFlow [3] developed by Google, PyTorch[4] developed by Facebook, CNTK[5] developed by Microsoft and MindSpore by HUAWEI.

At present, there are two execution modes of mainstream deep learning frameworks, namely static graph mode and dynamic graph mode. Static graph mode has high training performance but is difficult to debug. Although the dynamic graph mode is easier to debug than the static graph mode, it is difficult to execute efficiently. Currently, PyTorch is the most popular framework in academia. PyTorch is popular for its ease of programming and debugging, using both static and dynamic computational graphs. Programming in PyTorch is more dynamic, with users defining and executing graph nodes during execution. PyTorch can suffer from Python interpreter overhead. MindSpore provides a unified coding method for dynamic graphs and static graphs, which greatly increases the compatibility of static graphs and dynamic graphs. Users do not need to develop multiple sets of codes, and can switch the dynamic graph/static graph mode only by changing one line of code.

There are many studies evaluating various deep learning frameworks on different hardware. However, they usually only focus on the output results of the frameworks to compare these deep learning frameworks, rather than systematically explaining the deep reasons for the output. The reason behind these two frameworks is rarely discussed because, due to their completely different software stacks (including the way computation graphs are constructed, runtime scheduling), it is difficult to compare the two frameworks.

In this article, our purpose is to deeply compare the performance differences between PyTorch and MindSpore under a single GPU. To make the work as comprehensive as possible, we selected 3 very classic neural networks, including CNN, RNN, and Transformer, which cover the fields of computer vision and natural language processing.

2. BACKGROUND

2.1. Neural Networks and Deep Learning

The emergence of neural networks has enabled the rapid development of deep learning and has received extensive attention in the field of artificial intelligence (AI). Beginning with AlexNet [6], various DNN architectures (GoogLeNet [7], ResNet [8]) have emerged one after another in a short period of time, providing better feature detection and accuracy. Typically, a DNN structure consists of an input layer, an output layer, and multiple hidden layers. These layers can be viewed as a set of operations. Some frameworks use layer abstraction, while others and TensorFlow use operator abstraction. There are different types of layers for different applications and purposes, such as convolutional, pooling and activation layers for feature extraction in image classification, attention layers for information filtering in NLP, and LSTM layer. Combinations of layers can be explored to meet the needs of an application, such as ResNet, even when existing layers are considered.

The purpose of deep learning training is to find a suitable set of model parameters to minimize the loss function, which reflects the error between the predicted result of the sample and the ground truth label. The training process usually consists of millions of iterations, each of which involves two computationally intensive stages, forward and backward propagation. In forward propagation, the training samples are input to the input layer, and the weights and biases are added to calculate the output feature map as the input of the next layer. Finally, the loss is calculated by comparing the output to the ground truth labels of the output layer, ending the forward pass. Backpropagation starts from the output layer, traverses each layer in reverse, calculates the gradient of the parameters of each layer through the chain rule according to the loss value, and optimizes the parameters. There are many optimizers for backpropagation such as Stochastic Gradient Descent (SGD), Momentum and Adam. In general, the loss value gets

smaller and smaller as the number of iterations increases. Training ends when certain conditions are met, such as the loss value is less than a threshold, or the validation accuracy is above a threshold.

2.2. Deep Learning Framework

There are currently two mainstream deep learning frameworks: one is to construct a static graph before execution to define all operations and network structures, typically TensorFlow, which improves training at the expense of ease of use performance during the period; the other is the immediate execution of dynamic graph calculations, typically represented by PyTorch. By comparison, it can be found that dynamic graphs are more flexible and easier to debug, but at the expense of performance. Therefore, it is difficult for existing deep learning frameworks to meet the requirements of easy development and efficient execution at the same time. This article uses the new deep learning framework MindSpore, which provides a unified coding method for dynamic graphs and static graphs, which greatly increases the compatibility of static graphs and dynamic graphs. Users do not need to develop multiple sets of codes, just change one line of code to switch dynamic graphs. Graph/Static Graph mode. The framework aims to achieve three goals: easy development, efficient execution, and full scene coverage. MindSpore provides users with a Python programming paradigm. With automatic differentiation based on source code transformation, users can use native Python control syntax and other advanced APIs such as Tuple, List, and Lambda expressions. The work in this paper mainly studies the performance comparison between PyTorch and MindSpore.

3. EXPERIMENTAL METHODS

3.1. Workloads Selection

This article tests the main areas of deep learning in order to be as comprehensive as possible. The test work selected two deep learning fields of computer vision and natural language processing in deep learning, and also included the current mainstream neural network architecture.

3.1.1. Computer Vision

Computer vision is a field of artificial intelligence. The image is fed into a neural network, which is trained through a series of mathematical calculations. A trained neural network can classify and detect objects in pictures or videos. In recent years, neural networks have developed rapidly in the field of computer vision. In this paper, we have chosen GoogleNet.

3.1.2. Natural Language Processing

Natural language processing is a field of artificial intelligence that enables computers to read and correctly understand the meaning of human language. The human natural language is input into the neural network, and the neural network is trained through a series of mathematical operations. The trained neural network can understand human language. RNN is a good model for natural language processing. In this paper, we choose the LSTM[9] and BERT[10] models for comparison.

3.2. Unify the Implementation between PyTorch and MindSpore

The implementation of the same neural network between MindSpore and PyTorch may differ in some aspects, which affects training performance and fair comparison. Therefore, we try to unify

the implementations of MindSpore and PyTorch in order to provide a fair comparison. We give implementation methods from two aspects of model structure and hyperparameters. The model settings are shown in Table 1.

Table 1. The settings in MindSpore and PyTorch

Domain	Model	Key Layer	Batch size	Dataset	Framework
CV	GoogleNet	Conv	128	CIFAR-10	MindSpore
CV	GoogleNet	Conv	128	CIFAR-10	PyTorch
NLP	LSTM	LSTM	64	Aclimdb_v1	MindSpore
NLP	LSTM	LSTM	64	Aclimdb_v1	PyTorch
NLP	BERT	Embedding Full-connect	8*256	Cn-wiki-128	MindSpore
NLP	BERT	Embedding Full-connect	8*256	Cn-wiki-128	PyTorch

3.3. Get Accurate Training Speed

Training a neural network can take anywhere from weeks to months. Due to the iterative nature of deep learning training, we only sample a small segment of the entire training, effectively collecting training performance. However, the sampling period may vary from input to input. For models with fixed input lengths, such as CNNs, training can stabilize quickly, which means that the difference between iterations is very small. Therefore, we can collect accurate training scores in a short time. For models with variable input length, such as RNN, the training speed is different for each iteration due to the different input size. In this case, training epochs (traversing the entire dataset) are required for stable performance.

In addition, there is usually a construction phase at the beginning of training to construct the computational graph, allocate memory, and modify some parameters (i.e., the workspace size of different convolutional layers).

Only after this does the computation at each step show repetitive behavior, which can be used to represent precise performance. Next, we describe the method to obtain accurate training speed in these 3 models.

4. EVALUATION

4.1. Experimental Setup

In order to ensure that the hardware is as unified as possible during training, we chose Alibaba Cloud GPU server ecs.gn6e-c12g1.3xlarge, 12-core Intel CPU, 92G memory, NVIDIA v100, and ubuntu18.04 system.

4.2. Overall Training Performance Comparison

In this subsection, we first look into the comparison of overall training performance between MindSpore and PyTorch. The results are shown in Table 2.

Table 2. Overall training speed on MindSpore and PyTorch

Model	Time	Loss	Acc
GoogleNet_MS	126.87(m)	0.0016	93%
GoogleNet_PT	152(m)	0.0016	94.68%
LSTM_MS	1049(s)	0.12	84%
LSTM_PT	1154(s)	0.0057	83.95%
BERT_MS	610(h)	1.7	58.88%
BERT_PT	1147.5(h)	1.71	59.21%

First, the overall training performance of PyTorch and MindSpore under the NVIDIA platform is compared, and the results are shown in the table. It can be seen from the results that the overall performance gap between MindSpore and PyTorch is small. By analyzing the experimental data, it is found that MindSpore's training speed is fast, but its accuracy rate is lower than PyTorch, while PyTorch is just the opposite. PyTorch's training speed is slow, but its accuracy rate is high. In summary, the overall training performance of MindSpore and PyTorch on the NVIDIA platform is very similar.

MindSpore is a deep learning framework developed by HUAWEI. They have developed a matching deep learning computing card Ascend910 for MindSpore. This paper also uses Ascend910 to test the above deep learning model. The experimental data is shown in Table 3.

Table 3. Training speed on Ascend

Model	Time	Loss	Acc
GoogleNet	63.85(m)	0.0016	93.4%
LSTM	523(s)	0.12	85%
BERT	384(h)	1.7	58.90%

Through experiments, we found that the speed of training with Ascend910 is much faster than the speed of training the model with the NVIDIA platform, and the accuracy is similar to the accuracy of the model trained with the NVIDIA platform. To sum up, MindSpore's accuracy rate on Ascend910 is similar to that on NVIDIA platform, but the training speed is faster than NVIDIA platform.

Training performance is an important indicator of deep learning models, and inference performance is also an important indicator in the use of deep learning models, as shown in Table 4.

Table 4. Training speed on Ascend

Framework	Model	Hardware	images/sec
MindSpore	ResNet-50	V100	1490.2
PyTorch	ResNet-50	V100	856.5
MindSpore	ResNet-50	Ascend910	2115

Through the data, we found that during the inference process, the speed of MindSpore is faster than that of PyTorch when using the NVIDIA platform, and the speed of using Ascend910 is much faster than that of using the NVIDIA platform. To sum up, the use of Ascend910 prevails when the application of the model is the primary selection criterion.

5. CONCLUSION

The ultimate goal of this article is to help end users make an informed decision between how to choose two of the most popular deep learning frameworks: MindSpore and PyTorch, in single-GPU training. We systematically evaluate single-GPU training on MindSpore and PyTorch using 3 representative models. Through these comprehensive experiments, we provide insightful observations and recommendations for end users and system developers. First, we decompose the training process of a single GPU, showing that the training process is mainly consumed by GPU processing, which is mainly the execution time of the kernel. Therefore, the running speed of key layers plays a crucial role in single-GPU training. We then evaluate the performance of various models implemented with different key layers and present the trade-offs among them to provide reference for end users to choose various implementations in reality. Finally, we evaluate the performance impact of MindSpore and PyTorch in the dynamic graph case. The conclusion is that when deciding between MindSpore and PyTorch based on training speed, choose MindSpore, and when deciding between MindSpore and PyTorch based on accuracy, choose PyTorch. Choose MindSpore when the application of the model is the primary selection criterion.

ACKNOWLEDGEMENTS

Supported by Beijing City University in 2021 “the innovation and entrepreneurship training program for college students”

REFERENCES

- [1] Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, Toronto, 2017. 1–12
- [2] Jia Y, Shelhamer E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, 2014. 675–678
- [3] Abadi M, Barham P, Chen J, et al. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 2016. 265–283
- [4] Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: Proceedings of the Autodiff Workshop on NIPS, 2017
- [5] Seide F, Agarwal A. CNTK: Microsoft’s open-source deep-learning toolkit. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 2135–2135
- [6] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2012. 1097–1105
- [7] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1–9
- [8] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
- [9] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31.7 (2019): 1235-1270.
- [10] Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. 2018. ArXiv:1810.04805

INVESTIGATING CARGO LOSS IN LOGISTICS SYSTEMS USING LOW-COST IMPACT SENSORS

Prasang Gupta, Antoinette Young and Anand Rao

AI and Emerging Technologies, PwC, India

ABSTRACT

Cargo loss/damage is a very common problem faced by almost any business with a supply chain arm, leading to major problems like revenue loss and reputation tarnishing. This problem can be solved by employing an asset and impact tracking solution. This would be more practical and effective for high-cost cargo in comparison to low-cost cargo due to the high costs associated with the sensors and overall solution. In this study, we propose a low-cost solution architecture that is scalable, user-friendly, easy to adopt and is viable for a large range of cargo and logistics systems. Taking inspiration from a real-life use case we solved for a client, we also provide insights into the architecture as well as the design decisions that make this a reality.

KEYWORDS

Asset tracking, Logistics, Cargo loss, Cargo damage, Impact sensor, Accelerometer sensor, Low-cost solution, No code AEP (Application Enablement Platform).

1. INTRODUCTION

Amid the advent of globalisation, the transit of goods from one location to another is a quintessential part of any business with manufacturing or a supply chain arm. However, due to the sheer amount of logistics involved in global transportation which requires an array of different stakeholders, the handling of the actual package to be transported takes a back seat. In many cases, these packages are poorly handled or there is a hefty premium involved for ensuring proper handling of the package which makes the whole transportation process too costly for the parent business. In fact, The National Cargo Security Council (NCSC) estimates that the global financial impact of cargo loss exceeds \$50 billion annually [1], moreover 50% of domestic and international insurance claims are denied [2].

The financial impact of goods damaged during transit goes beyond the cost of replacing damaged cargo [3]. There are many other factors to be considered such as interruptions to the supply chain, higher insurance claim costs, loss of productivity due to filling out claims, tracking down return orders, repackaging and shipping replacements, etc. Damaged cargo can also damage relationships among the firm, vendors and clients and can also project a negative impact on an organizations' brand value leading to further loss of sales and market share.

This serious problem of damaged goods with dire consequences for any business is surprisingly straightforward to mitigate with some care and infrastructure in place. Cargo damage can be prevented if the right measures are taken. However, prevention requires information regarding the conditions that the cargo experiences while it is in transit [4]. Increasing the visibility in the transit section of the supply chain could allow the businesses to track the amount of cargo damage occurring [5] and provide valuable information relevant to mitigating this loss. One of

the possible methods is to use business frameworks and high-level risk analysis for the transit [6]. This also includes laying out operational guidance for the crew [7]. Another method is to collect the information regarding the impacts that the cargo faces while in transit as well as the geographical locations and the intensity of the impacts. This can be done by actively monitoring the cargo using sensors. We will discuss the latter in this study.

Tracking a shipment is a very common and important problem statement. There are several solutions that are present in the literature including using LoRa WAN technology [8], radio frequency identification (RFID) [9], or dynamic scheduling [10]. We explore a different strategy here and focus on capturing and tracking the impacts faced by the cargo [11] with the prime focus of keeping the solution cost effective and user friendly.

There are many types of sensors that can be used to measure impacts and vibrations which can be helpful in preventing damage to the cargo [12]. However, most of these sophisticated sensors cost upwards of a couple thousand dollars [13,14]. While this may be viable for high-value goods, this might not be cost-effective if the cost of the sensor is comparable to the cost of the cargo, which is going to be the case for most types of goods and businesses.

We discuss here a solution that we employed for one of our clients facing a very similar issue. The salient feature of the solution is that it can be scaled, it is generic enough to be extended to multiple problems and requires very low upfront capital investment, hence, is economically feasible to implement for most types of cargo. We will discuss the general solution architecture and design choices across the study, with short details and specifics about our client problem to give a flavour of a real working solution.

In the next few sections, we will first discuss the types of sensors that are viable for this kind of a problem along with the sensor that we used with a brief overview of its features. Then, we will move on to discuss the analytics behind converting the raw sensor values to meaningful information. Lastly, we will discuss how we can bring all of this together in a holistic solution architecture using an application enablement or a cloud platform.

2. SENSOR

We used a cost-effective accelerometer sensor to capture vibrational and/or impact data instead of sophisticated sensors in this study. This section starts off with a brief description of the accelerometer sensor. The next section details the different firmware-level changes that need to be made on the sensor to suit our specific class of problems. Finally, we will have a brief discussion on the features that were instrumental in the decision behind selecting the sensor.

2.1. Accelerometer Sensor

An accelerometer is a sensor which measures proper acceleration. Proper acceleration is the acceleration faced by a sensor in its own frame of reference. This implies that the acceleration recorded by the accelerometer while sitting statically would be 9.8 m/s^2 on the axis which is perpendicular to the ground and 0 on the other two axes. For e.g., the y axis of the accelerometer sensor in Figure 1 would register an acceleration of -9.8 m/s^2 while the x and the z axis would register 0 acceleration.

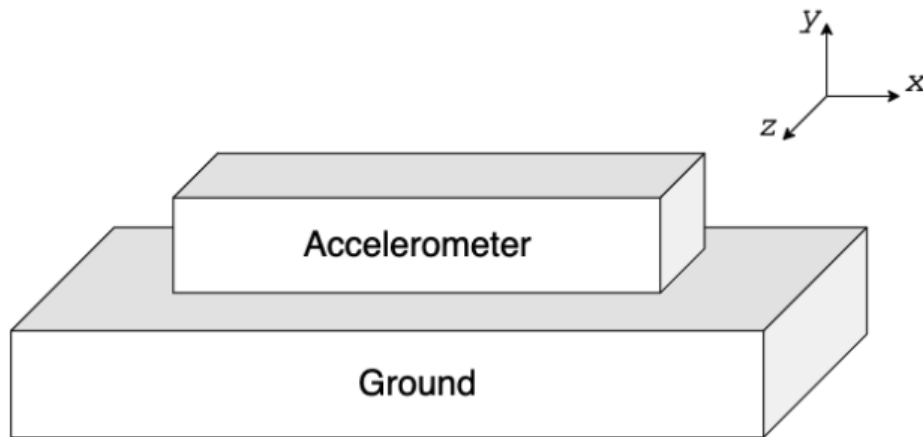


Figure 1. Accelerometer sensor placed with y axis perpendicular to the ground

The accelerometer sensors are usually configured to report the acceleration values multiple times within a second. The frequency of each sensor depends on the hardware capabilities of the sensor as well as how it is configured. Also, the maximum acceleration that can be recorded by the sensor before it is capped off, also varies between different configurations.

The accelerometer sensor that we used was housed in a multi-purpose device built by a location tracking equipment manufacturer. The device also contained a temperature sensor, a humidity sensor and GPS along with memory and communications support. This allowed the device to store collected data based on triggers and then flush it to the cloud whenever an internet connection was established. This connection establishment to the internet could be put on a schedule if needed. It also contains firmware which allows configuration of each of the sensors housed inside the device. We used this firmware to tweak the accelerometer which is discussed in the next section.

2.2. Firmware Changes

As explained earlier, this device collects and stores the accelerometer data multiple times within a second. The general use case does not benefit from getting instantaneous acceleration of the device, however it does by monitoring the sudden changes in acceleration which are characteristic of sudden jerks or an impact. This can be achieved by listening and recording all the values that cross a particular threshold. This threshold can be decided based on the fragility of the cargo. We optimised the threshold by performing several drop tests and calibrating the accelerometer raw values at the point where the drop was lethal to the package.

Another important aspect of the solution is to send the raw data collected by the sensors to the cloud periodically. Sending data to the cloud is a battery-intensive process, hence, it needs to be throttled to enable the device to keep running for long periods of time without a need to be recharged mid-transit. The cloud connection schedule of the device also needs to be optimised to confirm that the device remains functional throughout the transit and does not drain the battery, prompting the device to need charging. This is important from a logistics perspective as the lower the maintenance required for the device, the better.

For our case, these firmware changes allowed the device to complete the transit in its first test run. The battery life of the device can be extended from a few days to a year after changing the cloud connection frequency. This allows the implementation to work without any need of

recharging of the device till the end of a cycle. Adding to that, when the demo cycle using the device in a real case study was finished, the device was still at about 30 % of battery capacity even after being used for about 4 months.

2.3. Sensor Requirements

As we have already discussed, there are several specific requirements for our general use case. Hence, a device needs to be selected that checks all the boxes and requirements. At the same time, it is important that the sensor remains cost-effective. Here is a summary of all the requirements that the selected sensor should be able to fulfil:

- As we are assuming that our shipping is across country borders, it can also be safely assumed that it will cover both land and sea routes. Hence, there is a requirement for the device to be able to connect to the cloud and send data while roaming on any cellular network available in a region. Connections using BLE (Bluetooth Low Energy) and WiFi won't work because of the unavailability of such networks in remote areas
- The total shipping time would be of the order of weeks. The device should last the whole journey without the need to be recharged. This would confirm hassle-free logistics and provide another layer of robustness to the solution.
- The shipping route would cover multiple nations and hence, the connectivity options should be nation-agnostic
- The data to be sent to the cloud is of the order of KBs (raw accelerometer values with timestamp and location information) and hence, the device needs to have sufficient on-board memory and should be able to support this bandwidth while sending data to the cloud

Owing to these requirements, we chose the connectivity option as LTE-m/NB-IoT with a 2G/3G fallback for the device. The device was configured to use 3G network, if available, to increase the speed of transfer. However, in the case when no 3G network was found, it would look for a 2G network and communicate through that. This network choice confirmed that:

- The device may connect to the cloud anywhere without any need for an established network (like Wi-Fi or Bluetooth)
- The bandwidth supported by 2G/3G networks is enough for sending data of the order of KBs to the cloud
- This option would also work in many regulatory domains because of the abundant presence of 2G/3G towers as opposed to only 4G networks.
- Using a 2G/3G network also enabled connectivity due to existing roaming agreements between carriers

3. ANALYTICS

The analytics and the transformations on the raw accelerometer data are the core of the solution of making the cost-effective sensors perform as good as sophisticated sensors for our general use case. These transformations would ideally provide us with actionable insights regarding the different impacts, including both the frequency and the level of the impact, as well as the locations of these impacts. The locations would help us in pinpointing the exact arm of transit, and impact can be mitigated by changing the transporter for that arm or by increasing investment in packaging based on the levels of impact faced.

The raw sensor data is in the form of a collection of objects. Each of these objects contains a timestamp, current battery level, accelerometer readings for each axis and additionally, readings of other sensors present in the device, if any.

When data is flushed by the device, an array of these objects are uploaded to the cloud. All of these objects have the accelerometer value above the pre-decided configured threshold. The data can be accessed for running analytics either directly through a data pull from the cloud or by putting an API (Application Programming Interface) service on top of the data storage and requesting it with configured parameters thereafter.

The device may contain other sensors apart from the accelerometer like temperature, humidity etc. We can choose to either disregard all the other sensors or spin up a dashboard around them for more insights about the transit. These sensors could be used to add further insight on the environmental factors that have a direct effect on the cargo. These can then be switched on and off in the analytics section accordingly.

The timestamp and battery information sent by the device follows a certain pattern specified by the manufacturer and would be straightforward to work upon. However, the raw accelerometer values generally come in scaled quantities. For our case, it was between -512 and 512, with 0 specifying no acceleration on that axis. The drawback of this was there was no attached physical sense to these readings and the impact values can only be used for comparison purposes.

To generate physical meaning from these values, they need to be calibrated. This was done by proposing a function that converts the raw accelerometer values into a force value in some physical units. We have chosen the 'g-force' unit for our study. The function that we used in our study was

$$V = \lambda \sqrt{x^2 + y^2 + z^2}$$

where

V =Impact value in g force λ =Case specific constant x,y,z =Accelerometer raw values

Equation 1. Calibration equation to convert raw accelerometer values to g-force units

The case specific constant mentioned in Equation 1 was calculated by performing several experiments using the sensor. Most of these experiments involved dropping the sensor from different heights and noting down the impact numbers registered by the accelerometer and then back calculating the constant value. These tests also helped in identifying the threshold g-force value that could be considered dangerous for the equipment to be tracked. This threshold would vary depending on the packaging and the fragility of the cargo.

4. SOLUTION ARCHITECTURE

Having all the pieces of the problem solved, this section deals with bringing them all together in a coherent architecture. There are several ways to design this. It could be an on-premises system, a cloud-based system or a hybrid between the two. Also, any platform that supports API requests and dashboarding can be used for creating the architecture. The architecture for the whole solution can be divided into 4 distinct components: hardware layer, data layer, ingestion engine, and presentation layer. This can be visualized in Figure 2. We will discuss each of these components in greater detail.

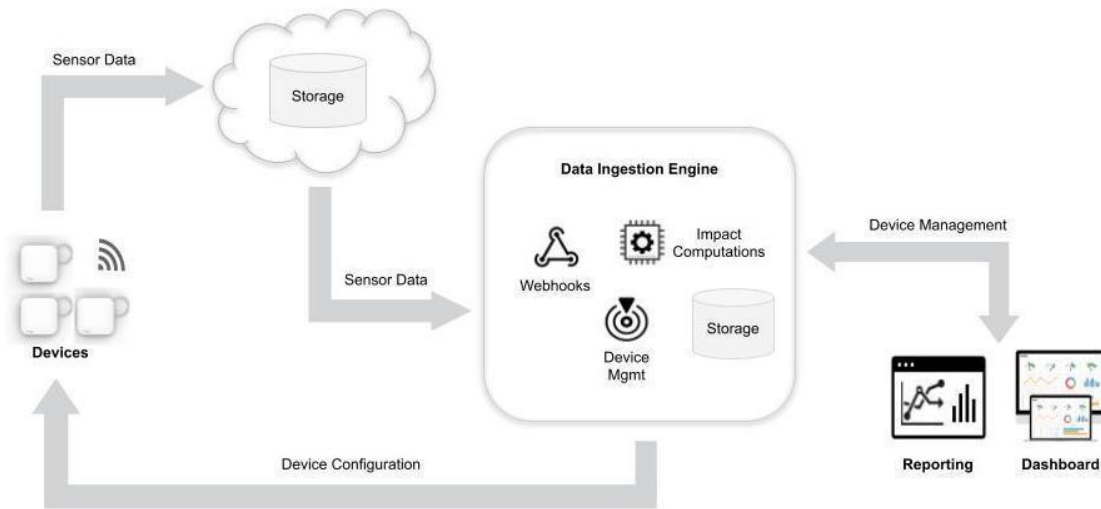


Figure 2. Overall Solution Architecture

4.1. Hardware Layer

The hardware layer consists of all the sensors required and the housing for the same. The purpose of this layer is to capture the values recorded by these sensors and store it in memory. Another responsibility of this layer is to connect to the cloud on a scheduled basis and flush the captured data to the cloud. This can be used either for analysis purposes or for logging and historical data collection if the analysis is done at the edge.

For our use case, we used an accelerometer housed in a pre-built key-sized device. The device also had the capability to connect with the cloud and flush its data stored in on-board memory. We configured this according to our need and that concluded our hardware layer solution.

4.2. Data Layer

The data layer consists of all the data that is being stored, either for archival or for analysis purposes. The responsibility of this layer is to manage all the data that is generated by the sensors and provide it on an on-demand basis.

For our use case, we configured the device to send the data to the cloud which in turn gets fed to a database. An API was built on top of that for retrieval based on timestamp range and device ID. This API was kept secure, and data was retrieved from this using proper keys for further analysis of the raw data stored.

4.3. Ingestion Engine

The ingestion engine consists of all components that are needed to convert the raw data into meaningful insights. This component includes compute engines, database, and ETL (Extract-Transform-Load) functions etc. The responsibility of this layer is to fetch the raw data from the data layer and make the data ready to be published by the presentation layer.

For our use case, we created an ETL engine and a database to store the processed input. The ETL engine was configured to get the data from the data layer API in short batches and update the processed database. This layer was hosted on a no-code AEP (Application Enablement Platform).

4.4. Presentation Layer

The presentation layer consists of all the components that the end user would directly interact with. This includes all dashboards and websites generated for the user's perusal. The responsibility of this layer is to fetch the processed data/insights from the ingestion engine and publish it in meaningful ways on a dashboard. This includes plots, buttons, colour coded depictions etc.

For our use case, we generated a dashboard and hosted it online. The dashboard generated a live time series plot of all the impacts faced by the device, which it fetched directly from the processed database. This plot was also configured to include the current general location of the device, hence enabling the user to ascertain the regions where the cargo is facing stronger impacts. The dashboard also contained other device-level information such as current battery level, last known location and the last connection timestamp with the device. A snapshot of the dashboard can be seen in Figure 3. This layer was also hosted on the same no-code AEP.

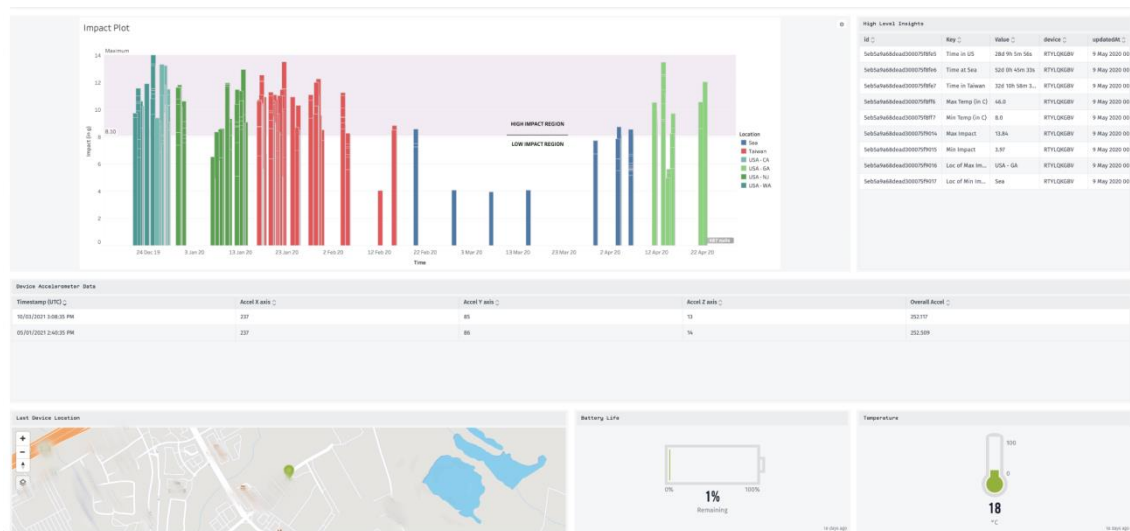


Figure 3. A snapshot of the dashboard published

5. RESULTS AND DISCUSSION

In the previous sections, we have mentioned and discussed in detail a method that can be employed for easy asset and impact tracking. We have also covered the key design aspects involved along with general guidance for making these decisions and a brief overview of the pros and cons of each, with respect to our particular use case. We will continue this discussion here and highlight the salient points of the solution developed and how it was helpful for the client.

The live dashboard generated for the client, as seen in Figure 3 was hosted online giving the client easy access for tracking all their current shipments in the same place. Apart from all the fields provided regarding the general information of the device, specialised impact plots were created highlighting the impact faced by the device in different legs of its journey along with the

demarcation of high/low impact. This was used to ascertain the regions where it was most probable for the cargo to get damaged. This plot can be seen in Figure 4.

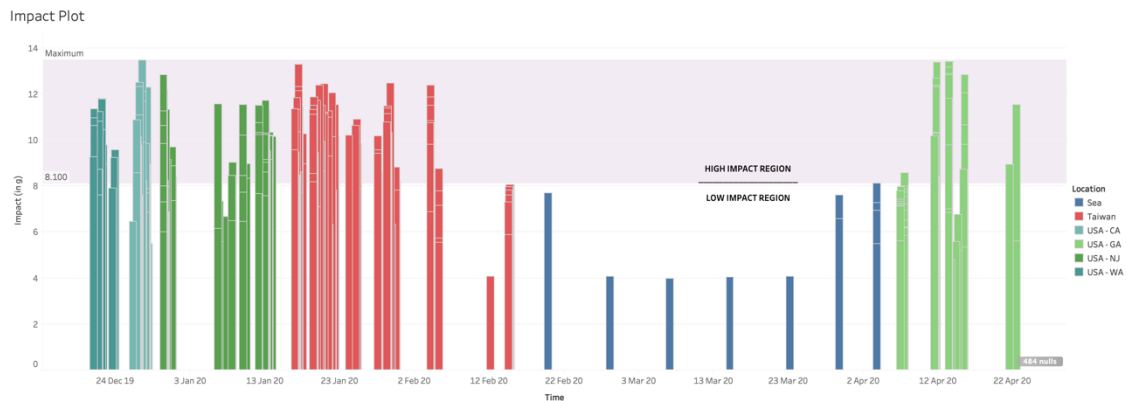


Figure 4. A snapshot of the impact plot generated

The plot generated in Figure 4 gave invaluable insights into the journey of the shipment from the manufacturer to the warehouse. It was concluded from the charts that most of the high impacts faced by the device were on land routes (red bars on the left are the impact readings when the shipment was shipped from the manufacturer and green bars on the right are the impact readings when the cargo was transported from the docks to the warehouse) rather than on sea routes (blue bars represent cargo travelling on container ship). This provided transparency to the client on their whole supply chain, creating value by assisting them in making decisions to reduce cargo damage.

Overall, this solution was built keeping the ease of logistics in mind, which included one-time use devices with no hassle of recovering them from cargo, a device that would not require charging within the transit period, and ease of placing the device on the cargo with no special instructions/handling directions etc. This proved to be very useful in terms of the viability and ease of adoption and scalability of the solution.

There are certain limitations to this type of architecture in the logistics space. Most importantly, this creates issues when there is a need for recovery of the tracking devices as this is not covered. Also, tuning frequency of the data collection and other device parameters needs some prior experience in the IoT domain to get the optimal values as recreating real-life situations is not possible. Another limitation of this method is the long feedback cycle as the parameters can only be optimised after a single shipment is complete. Hence, getting the desired parameters could take 2-3 shipment cycles, assuming that the cycle remains the same every time.

6. CONCLUSION

Damaged or lost cargo can have a negative impact on every business with a supply chain arm. This impact can be both short term like product loss, replacement inventory, time lost in filing claims etc. and long term like loss of clients or market share, loss of trust in the brand etc. As discussed, a major share of damage or loss of cargo happens during its transportation.

Reducing in-transit damage by leveraging the latest technologies for monitoring impacts and vibrations can offer several benefits to the business and help them in the long run. Some of these benefits are:

- Real-time tracking of impact, vibrations etc. to the cargo increases visibility through the transit route and helps determine where and when damage is occurring even when out of coverage.
- Monitoring provides data that can be used to help prevent damage in the future by using proper packaging for the cargo.
- Understanding the trends in damage occurrence, delays, and other issues can help in improving performance leading to better customer service and satisfaction

This work, with careful calibration and design choices, can further be expanded from impact tracking to much more sophisticated vibration tracking. This would open up arrays of opportunity for this system to be implemented. Also, other sensors can be used in conjunction with the accelerometer sensor to solve multiple complex problems.

REFERENCES

- [1] T. Hayes, "The full cost of cargo losses," <https://www.inboundlogistics.com/cms/article/the-full-cost-of-cargo-losses/>, 2004, [Online; accessed 21-Oct-2021].
- [2] J. Paul Dittman, "Will you be ready when a loss happens to you?", <https://upscapital.com/wp-content/themes/upscapital-bren/assets/media/Loss-whitepaper.pdf>, 2015, [Online; accessed 21-Oct-2021].
- [3] Liu, Minhui, Mandyam M. Srinivasan, and Nana Vepkhvadze. "What is the value of real-time shipment tracking information?." *IETTransactions* 41.12 (2009): 1019-1034.
- [4] P.-J. Wu, M.-C. Chen, and C.-K. Tsau, "The data-driven analytics for investigating cargo loss in logistics systems," *International Journal of Physical Distribution & Logistics Management*, 2017.
- [5] A. T. Yu, "Examination of bulk cargo loss in transit," *Journal of the National Academy of Forensic Engineers*, vol. 3, no. 1, 1986.
- [6] H.-Z. Zhang, C.-M. Hsieh, Y.-L. Luo, and M.-C. Chiu, "An investigation of cross-border E-commerce logistics and develop strategies through SCCOM framework and logistic service risk analysis," *Transdisciplinary Engineering: A Paradigm Shift*, vol. 5, pp. 102–113, 2017.
- [7] V. Shigunov, O. E. Moctar, and H. Rathje, "Operational guidance for prevention of cargo loss and damage on container ships," *Ship Technology Research*, vol. 57, no. 1, pp. 8–25, 2010.
- [8] F. Flammini, A. Gaglione, D. Tokody and D. Dohrilovic, "LoRa WAN Roaming for Intelligent Shipment Tracking," 2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), 2020, pp. 01-02, doi: 10.1109/GCAIoT51063.2020.9345843.
- [9] C. Hillbrand and R. Schoech, "Shipment Localization Kit: An Automated Approach for Tracking and Tracing General Cargo," International Conference on the Management of Mobile Business (ICMB 2007), 2007, pp. 46-46, doi: 10.1109/ICMB.2007.58.
- [10] Brewer, A., Sloan, N. & Landers, T.L. Intelligent tracking in manufacturing. *Journal of Intelligent Manufacturing* 10, 245–250 (1999). <https://doi.org/10.1023/A:1008995707211>
- [11] Shamsuzzoha, A. H. M., and Petri T. Helo. "Real-time tracking and tracing system: Potentials for the logistics network." In Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management (IEOM), pp. 22-24. 2011.
- [12] S. Hanly, "What Really Happens to Your Package During Shipment?," https://blog.endaq.com/transportation-vibration-monitoring-what-really-happens-during-shipment?utm_source=PwC&utm_medium=link&utm_campaign=visitor&utm_content=link-from-pwc-article-to-package-during-shipment-blog, 2021, [Online; accessed 21-Oct-2021].
- [13] "enDAQ Sensors Shock, Vibration & Environmental Sensors," https://endaq.com/collections/endaq-shock-recorders-vibration-data-logger-sensors?utm_source=PwC&utm_medium=link&utm_campaign=visitor&utm_content=link-from-pwc-article-to-endaq-pricelist, 2021, [Online; accessed 21-Oct-2021].
- [14] "Impact Sensors, Impact Monitors & Shock Sensors," <https://spotsee.io/impact>, 2021, [Online; accessed 21-Oct-2021].

AUTHORS

Prasang Gupta is an Experienced Associate in PwC's AI and Emerging Technologies team. He has 2 years of experience in solving complex real-world problems with cutting edge ML and AI inspired innovative solutions. He earned his Master of Technology and Bachelor of Technology in Chemical Engineering from Indian Institute of Technology Kanpur.



Antoinette Young is a manager in PwC's AI and Emerging Technologies team. In this role she does research and develops prototypes focused on Internet of Things (IoT) technologies. Antoinette is passionate about all things robotics. She earned a Master of Science in Information Technology from Nova Southeastern University and a Bachelor of Science in Computer Science from Florida Atlantic University.



Anand Rao is a principal in PwC's AI and Emerging Technologies team and the global AI lead with over 30 years of industry and consulting experience, helping senior executives structure, solve and manage critical issues facing their organizations. He holds a MSc (Tech) in Computer Science from Birla Institute of Technology and Science in India and PhD in Artificial Intelligence from University of Sydney, where he was awarded the University Postgraduate Research Award. also awarded an MBA from Melbourne Business School with Distinction.



PwC refers to the US member firm, and may sometimes refer to the PwC network. Each member firm is a separate legal entity. Please see www.pwc.com/structure for further details. This content is for general information purposes only, and should not be used as a substitute for consultation with professional advisors.

UNDERSTANDING THE EFFECT OF IoT ADOPTION ON THE BEHAVIOR OF FIRMS: AN AGENT-BASED MODEL

Riccardo Occa and Francesco Bertolotti

LIUC – Università Cattaneo, Corso G. Matteotti 22, Castellanza (VA), Italy

ABSTRACT

In the context of increasing diffusion of technologies related to Industry 4.0 and the Internet of Things in particular, we have developed an agent-based model to simulate the effect of IoT diffusion in companies and verify potential benefits and risks.

The model analyses how firms react to the spread of IoT in the market by analysing its effects on firms' pricing and quality strategies, its impact on variable production costs, and the long-term survival rate of firms.

The model shows how IoT diffusion has the potential to influence the market by supporting both quality and cost improvements.

The results of the model also confirm the potential for significant benefits for businesses, suggesting the opportunity to support the introduction and application of IoT, and clearly show how the use of IoT can be a key strategic choice in competitive market contexts focused on cost strategies to increase business performance and prospects.

KEYWORDS

IoT, agent-based modelling, simulation, adoption, risk, blockchain.

1. INTRODUCTION

The Internet of things has been one of the key components of Industry 4.0 and has seen an increasing spread over the last 10 years. For example, between 2010 and 2017, the number of IoT connections per 100 inhabitants worldwide increased from 2.5 to 14 [1].

The term IoT was first used by Kevin Ashton in 1999 in connection with his work at the Procter & Gamble Company on the potential of RFID [2].

Since then, the interest of the scientific world in the topic of IoT has never ceased. Instead, publications have been explosive growth focusing on IoT issues, exploring its different components, functionalities, and applications. Figure 1 represents the growth of papers that include IoT or the Internet of Things in the title within the SCOPUS database.

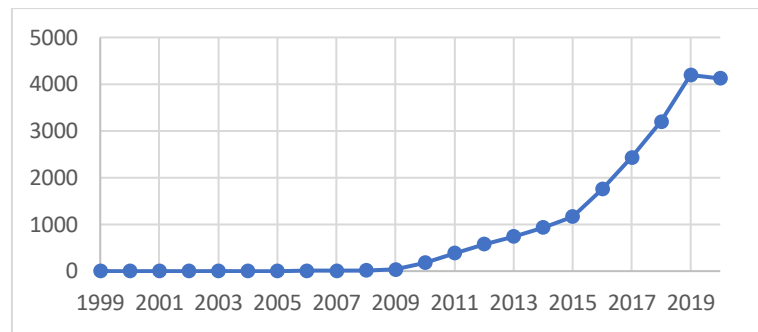


Figure 1. Documents published by year with IoT or Internet of things in the title .Source Scopus database

The scope of the IoT has rapidly expanded beyond manufacturing plants, production, and assembly lines or warehouses, involving many different sectors as shown in figure 2. Among the most widespread applications we can find today:

- Applications related to smart grids and the monitoring and control of electricity transmission generation and consumption systems [3]
- Applications related to the concept of smart cities and smart buildings, equipped with control systems that allow more efficient management of resources, spaces, and the introduction of new services and benefits for the people who live in these places [4]
- The theme of home automation and smart homes, with the possibility, for example, of remotely controlling household appliances and functions in the home [5]
- Smart logistics, with the development of real-time tracking or constant monitoring of product storage characteristics [6]
- Smart mobility, with new services dedicated to the world of the car, such as predictive maintenance and control of vehicle parameters, as well as more precise navigation tools that also allow traffic optimization, leading to autonomous driving [7]
- The development of smart offices, where technology can improve the productivity of the environment and simplify the activities of staff, improving the light and climate conditions of the environment or allowing the introduction of new safety measures.[8]
- The range of services linked to smartphones and wearable devices, which allows applications to monitor one's physical condition or the development of services linked to the habits and behavior of individuals.[9]



Figure 2. IoT applications area Source: [2]

IoT has therefore found wide application in manufacturing plants, through the analysis of machinery to enable predictive maintenance, the control of production processes to eliminate defects and waste, the possibility of developing new products and services, and, in general, a greater possibility of controlling and monitoring activities.

However, the introduction of the IoT has also led to the emergence of new risks for businesses, especially of a cyber nature. These risks are linked to several aspects of IoT technologies, including the need to protect and store large amounts of sensitive data and information, the creation of numerous new access points to corporate networks that can be accessed by malicious attackers, and the growth in the dependence of production activities on information and IT systems, which increases the magnitude of damage resulting from their malfunctioning.

Already in 2015, for example, [10] showed that in the previous two years, the growth in the number of connected devices worldwide was followed by an exponential growth in DDoS attacks, measured in bandwidth used.

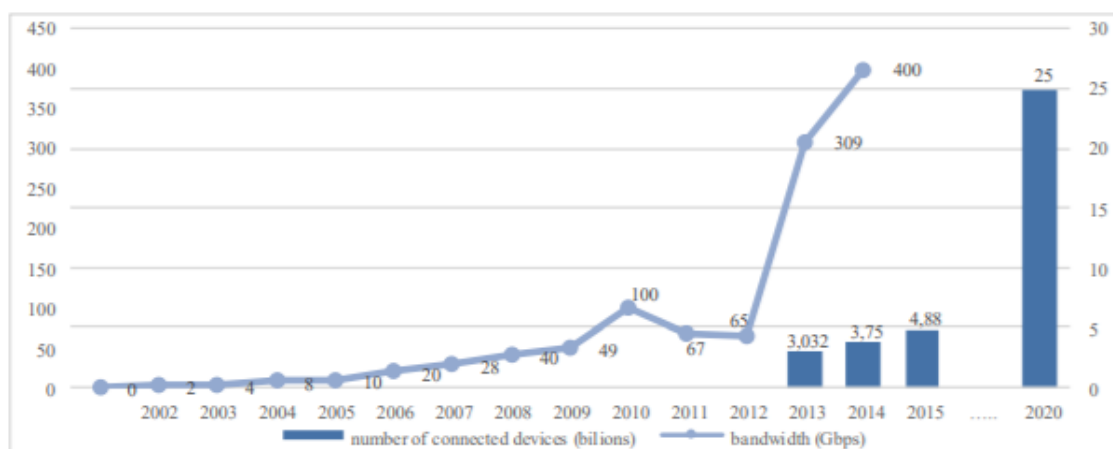


Figure 3. Connected devices and DDoS attack bandwidth. Source: [10]

Cyber security risks have been one of the most limiting factors in the spread and adoption of IoT [11]. For this reason, in recent years there has been a lot of research dedicated to solving and mitigating these issues. Among the methods for solving these problems, one of the possible solutions proposed was the adoption of Blockchain technology, resulting in the development of various solutions and proposals that combine the benefits of the two technologies [12].

An agent-based modeling (ABM) was developed to study these phenomena. ABM is a simulation technique in which a system is modelled by creating an assortment of autonomous decision-making entities which stands for the entity of the modelled system [13]. While an ABM is not the only available technique, but it is well fitted to model and simulate complex systems in which individuals take heterogeneous decisions following rules and observe the macro behavior of the system.

This work presents a simple agent-based model of a market of productive firms, in which each company competes. Firms have a given cost structure and can vary their strategies, to increase or decrease the quality and the price of the product. Each firm has the possibility to invest in the adoption of IoT technology, which can reduce the variable cost of production (e.g., higher efficiency), but allows for IoT attacks. Consequently, the decision implies a given amount of uncertainty, an expected benefit, and potential damages.

The purpose of the model is twofold. On the one side, it attempts to investigate the relationship between the IoT risk conditions, and the strategies adopted by firms. On the other, it finds a relationship between the adoption of IoT and the probability of survival in the market, especially when the market prefers low-cost products to high quality ones.

The structure of the paper will proceed as follows:

- section 2 will describe the current state of IoT technologies, with a focus on adoption and deployment and the characteristics of the Industrial Internet of Things. The security issues of IoT will then be presented.
- Section 3 will describe the characteristics of the agent based model adopted. The mathematical relations at the base will be presented and the assumptions of the simulations will be explained.
- Sections 4 and 5 will show the results obtained from the model and the discussions and conclusions derived from them.
- Section 6 will include proposals for future developments and additions to the proposed model.

2. BACKGROUND

2.1. IoT Adoption and Diffusion

As mentioned above, the growth of the IoT in recent years has been remarkable, However, this growth has not always been uniform across countries or economic sectors, and sometimes different applications have shown different peculiarities and characteristics.

Looking at Table 1, we can see that the number of IoT connections varies in different countries, with a higher diffusion in the EU, USA, and China than in other areas of the world.

Table 1. IoT connections per 100 inhabitants in different countries. Source: [1]

Country	2010	2011	2012	2013	2014	2015	2016	2017
China	0.8	1.4	2.4	3.6	4.9	7.3	12.2	24.0
France	4.1	5.5	7.3	10.6	12.6	16.0	17.1	22.5
Germany	2.9	3.7	4.9	6.4	8.4	11.2	13.7	16.6
Japan	3.4	4.7	6.1	7.2	8.9	10.1	12.1	13.8
Sweden	24.1	31.8	41.3	53.7	63.2	68.8	88.0	105.8
United Kingdom	5.0	6.4	7.9	8.7	10.4	12.4	15.3	17.6
United States	5.8	7.6	9.4	11.6	13.9	17.3	21.0	27.5
EU countries	4.9	8.7	11.5	14.3	16.9	19.6	23.3	27.7
OECD countries	4.9	8.0	10.4	12.9	15.3	17.8	21.2	25.4
Non-OECD c.	0.6	1.0	1.5	2.0	2.6	3.2	4.1	5.2
All countries	2.5	4.1	5.4	6.7	8.1	9.6	11.5	14.0

However, even within these areas, we can find differences.

If we look at the data on IoT adoption in businesses in Italy, [14] we can see that in 2019 73% of large companies will have started IoT-related projects, compared to 29% of SMEs.

However, in Germany, studies such as [15] have shown that company size is not a factor in IoT adoption.

Some countries have specificities related to the different levels of infrastructure available, e.g. in India, one of the main determinants of IoT adoption is the availability of adequate Internet connections [16].

However, numerous studies and analyses have made it possible to observe certain basic limitation factors common in different sectors or countries. These limiting factors include privacy and security issues, lack of defined standards, and issues related to lack of organizational support [17][18].

A final element to consider regarding factors that may limit the adoption of IoT is the cost of implementation. There is no clear position in the scientific world about the actual impact of costs in IoT adoption.

Some studies, such as [19], do not believe that the cost of implementing IoT technology is a relevant factor in influencing the adoption rate. Other studies, however, such as [20], have identified cost as a limiting factor, especially in SMEs.

2.2. The Industrial Internet of Things

The applications of the Industrial Internet of Things are varied and strongly influenced by the sector in which different industries operate and their supply chains.

IIoT in companies is generally able to lead to significant performance gains. At a general level, [1] has shown that within a country or an increase in IoT device connections corresponds to an increase in total factor productivity (TFP).

At the individual enterprise and application level, productivity gains are found in several IIoT applications.

The use of IoT for the introduction of IoT and Edge Computing-based Manufacturing, for example, allows for the development of decentralized mass production models, where the reduced operation completion time required allows for an increase in operational performance resulting in increased productivity [21].

Another area of application is monitoring and control, where the ability to track data at every point in production, including storage parameters during transport activities, allows for a minimization of waste and a reduction in the time required for machine and product control activities [21]. Monitoring can also be used to introduce predictive maintenance mechanisms to optimize plant and machinery set-ups and limit the occurrence of faults and waste in production. It should also not be overlooked that the systematic collection and analysis of production data can be the basis for the development of further simulations and analyses for the introduction of further productions or for the introduction of innovations aimed at improving existing productions [22].

Two other areas of IoT application that can often be found together are robotics and localization. Concerning industrial robots, in addition to the aforementioned issue of predictive maintenance, which can also be applied here, a new possibility opened up by the use of IoT is that of reactive replanning. The introduction of robots in production entails the development of more dynamic and reactive environments, which increases productivity but also increases the risk of errors or accidents when a predetermined action of a robot collides with a different situation or unexpected human behavior. Reactive replanning allows, thanks to the constant analysis of movements and the introduction of an automatic supervision mechanism, to react to unforeseen situations, or to modify the behavior of the machines in a new pattern more efficiently than the previous one when the opportunity arises. This makes it possible to increase safety and optimize the operation of the production system [23].

The localization of robots is an active part of this process, but it also finds great application in the automation of tasks performed by moving automatic components. A classic example is Amazon's Kiva robots, used in the handling of goods in warehouses, which enable major performance improvements, with cases of up to 50% more goods being stored in the same space and significant gains in efficiency [24].

2.3. The IoT Security Risks

When talking about IoT cybersecurity issues or naming cyber-attacks, we are not referring to a single issue that needs to be addressed. There are many different types of actions that can damage an IoT system or its user.

[25] have proposed the classification of threats shown in Figure 4, which distinguishes the risks into 4 different types; these categories are found in most of the proposed classifications:

Physical attacks: involve a reduction in the capacity of a network of IoT systems or part of it, usually requiring proximity to the systems by the implementer. The consequences of these attacks can include battery depletion, with the consequent inability to continue activities, as in sleep denial attacks, or the theft of information, as in the case of fake node injection or side-channel attacks, or the interruption of network operation, as in tampering or permanent denial of service [25] [26]. Another possible source of this type of attack can be the ones based on social

engineering techniques, which involve manipulating users to obtain sensitive information or access the network [26].

Network attacks: involve using the IoT network to create damage. Here again, the purposes can be varied, from intercepting sensitive information (sinkhole or man in the middle attacks) to altering or deleting data and information (spoofing or unauthorized access), to disabling the network or part of it (routing information attacks or Distributed Denial of Service attacks) [25][26].

Software attacks: are based on the use of software, Trojans, or malware to overcome the protective measures of IoT networks and gain access to information or the ability to modify and delete data. The targets of these attacks may include reaching and infecting elements such as data centers and cloud centers [25] [26].

Data attacks: relate to three types of cyber damage, data manipulation in data centers, unauthorized access to data centers and unauthorized disclosure and dissemination of personal data [25].

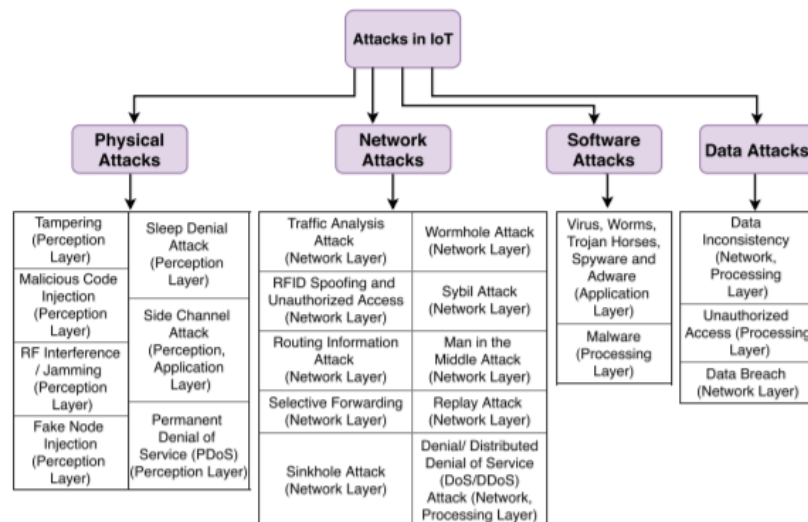


Figure 4. IoT security attacks classification Source: [25]

Different types of cyber-attacks require different and specific countermeasures. In addition to specific solutions, however, the growth of the IoT has stimulated the search for models and architectures that can resist different types of threats more effectively [27] [28]

Among these architectures, several are based on blockchain technology, which thanks to several of its characteristics such as the non-repudiation of transactions, decentralization, or the difficulty of modifying the transactions entered promises to provide better performance in terms of security [29].

Blockchain architectures for IoT are often multi-layer architectures because the use of blockchain can be costly in terms of energy and computing power. To this purpose, proposals envisage the use of "regional" nodes [30], which transfer the most onerous tasks to a smaller number of dedicated components, in order to guarantee normal performance for the sensors and elements that compose the IoT infrastructure.

Other architectures focus on particular aspects or processes related to IoT networks, [31] for example proposes a blockchain architecture to manage authorizations and access and ciphration keys. These are just a few examples of the many models developed and proposed.

3. AGENT-BASED MODEL

3.1. Model Description

We introduce here the model underlying the simulation results displayed in the succeeding section. The model has an exploratory intent. Consequently, we used a few simplifications. This was necessary for a twofold reason. Firstly, the more abstract is the model, the more general could be the results. Secondly, we aim at identifying and study a single phenomenon. A simpler model could be more suited to address the causal effect of a parameter variation connected to the IoT effect.

In the model, there is a single breed of agents, the firms. Firms can take two kinds of actions: they can operate on the market, or they can perform investments in IoT. Firms cannot communicate with each other, they only interact indirectly through their interactions with the market and their results. Consequently, the topology of the ABM can be described as a star network, with a single central entity (the market) with whom the agents interact. The dynamic of the market is very simple. Especially, we supposed that firms are homogeneous related to their size, produce and sell only one kind of product and their confirms are supposed to have only two kinds of costs: fixed costs and variable costs. Fixed costs are the same for each firm, and do not change in time. Variable costs depend linearly on three elements: the quality of the produced product, the number of products sold, and the efficiency of the production process, moderated by the level of adoption of IoT technologies and their influence on the variable costs. Mathematically, they are defined as:

$$vc = q \cdot c_q \cdot (1 - e) \cdot (1 - i_i \cdot i_s)$$

with vc variable cost of production, q product quality, c_q cost of quality, e efficiency of the production process, i_i IoT technologies adoption, and i_s IoT technologies savings.

Therefore, a firm is sustainable only when the difference between turnover and variable cost is greater than fixed costs. Hence, a successful strategy is any strategy that allows a firm to obtain this target. A strategy is made by a couple of elements: product price and product quality. In the model, it is assumed that both can be changed at any time. Moreover, in the market, a product is sold in a single distribution channel at the same price.

At each time step, the market has a certain demand for the product sold by the modelled firm. The total demand is sampled at each time step from a random normal distribution with a fixed mean and variance. The unity of measure of the demand is the amount of goods that the market requires. After the demand is generated, an allocation process takes place. First, all the quality and prices of each firm are collected and normalized. The prices are normalized inversely so that the minimum price corresponded to the value 1 and the maximum price with the value 0. It models the preference of customers for cheaper products. Second, a product success index is computed as follows:

$$\text{psi} = q_n \cdot \text{pm} + p_n \cdot (1 - \text{pm})$$

with psi product success index, q_n normalized quality, p_n normalized price and pm preference of the market (0 if it considers only price, 1 otherwise). This formula has two implications. First, higher normalized values of price and quality can bring to higher product success index. Second, the final value depends on the specific features of the simulated market. The market share of each firm i at each time step is allocated according to the following rule:

$$ms_i = \frac{psi_i}{\sum_i psi_i}$$

This allocation process is possible only assuming that every customer can purchase the product, and there are not any logistic limitations.

The decision making of agents is simple. While competing in the market, firms can perform two actions: the production (and automatically selling) of products and the change of strategy. The production level at each time step coincides with the demands coming from the market to the specific firms. In the model, no firm has a production capacity limitation, and the presence of time delay between the production moment and the availability of a product on the market is ignored because we considered these two factors not relevant for the risk preference adaptation of firms regarding IoT risks. In this first phase, the cash level of a firm is updated according to the profit. If the profit is positive, the cash level increases; otherwise, it decreases. If the cash level of a firm goes to 0, the firm fails, and it is removed from the simulation. Regarding the definition of strategies, agents behave naively towards the market, and they do not have any assumption related to the reason why a strategy is successful or not. When an agent is having a negative result (e.g., a negative profit) for more than a certain amount of time steps (defined by a parameter), it can change strategy while picking a new couple of product quality and product price at the same time. The new values are sampled from a random continuous uniform distribution. The changing in pricing strategy does not directly affect the cost of the product but the value of the markup, which affects the selling price as the following equation describes:

$$p = c \cdot (1 + m \cdot \mu)$$

with p product price, c product cost, m markup and μ price strategy.

In this model, firms can decide or not to adopt an IoT technology into their production process. The effect of the introduction of IoT relates to production efficiency. The IoT level of a firm is defined by a real number between 0 and 1. The higher is the IoT level of a firm, the lower are the variable cost of production of a product. It means that it will be more easily sustainable (because of the lower variable costs) and more successful on the market (because a reduction in the production costs implies a reduction of the selling price, which is computed with a markup). So, the level of the benefit is regulated by a specific parameter. Nevertheless, implementing IoT makes it possible to be a target of IoT attacks. The effect of an IoT attack is to take from the firm a given amount of money, reducing the cash level. The amount of money is computed sampling from a power law distribution, which exponent is a parameter of the model. The sampled value (between 0 and 1) is later multiplied for the last turnover of the firm, to give the appropriate magnitude to the IoT issue, because in real-world the monetary cost of an attack is related to the size of a firm. Hence, a firm can decide to invest or not in IoT technologies, according to their IoT adoption inclination, which stands for their profile of risk. The higher is their inclination, the more likely is for a firm to invest in IoT technology.

The scheduling of the model proceeds as follows. First, the agents' list is shuffled, so that the order in which firms are called changes at each simulation. Second, each agent updates the quality and the selling price (which derives from the production cost) according to their strategy.

So, if in the previous time step the strategy did not change, quality and selling price remains the same. Third, the market allocates the demand to different firms, and their financial status is updated with the profit. Fourth, the model removes the firms with a cash level below 0. Fifth, if firms had a negative performance in terms of profit in the last periods (the exact number is a parameter of the model) and did not change the strategy recently (as well, the minimum number of periods between two changes is a parameter of the model), it can change strategy with a given probability. Sixth, firms can invest or not in IoT technologies. Finally, firms that employ IoT technologies can experience informatic attacks. Figure 5 shows the scheduling of the model.

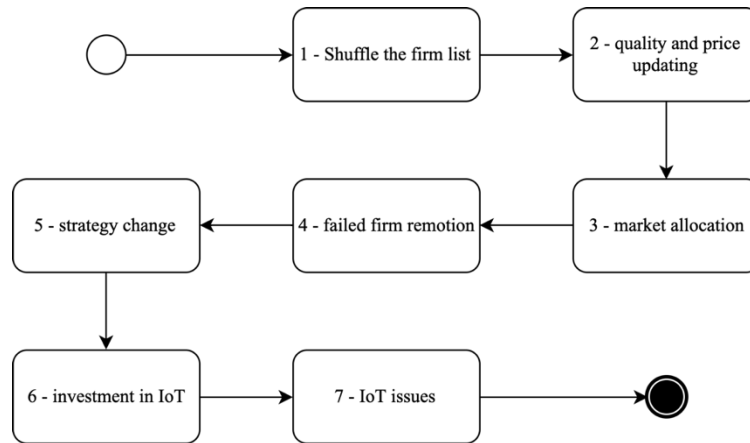


Figure 5. Scheduling of proposed model

The model is developed using Python 3.8 as programming language, without employing any specific framework for agent-based modelling, in an Anaconda environment.

3.2. Model Exploration

The results are achieved by simulating the model 20'000 times, varying the parameters. The purpose of this methodology, called grid sampling, is to identify the effect of each parameter on the outcomes, and consequently identify some specific relationships between the operating conditions and the outcome of the model. Under this point of view, it is possible to describe the model exploration as a black box. The inputs are the parameters variation, while the output the resulting values of the model. Figure 6 describes it.

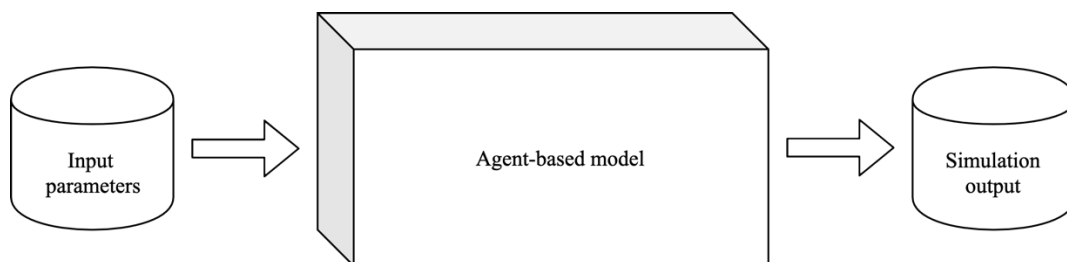


Figure 6. Black box concept application in the model

The parameters shuffled during the grid sampling are observed in Table 2, while Table 3 resumes the outcome of the simulation, that we employed to analyze the results.

Table 1. Parameter shuffled during simulations

Parameters	Meaning
init_firms	Number of initial firms in the market
pref_mkt	Preference of the market in terms of quality or price
cf_act	Fixed costs of activity of firms
q_cost	Variable cost for each unity of quality of the product
max_mkup	Maximum markup (price on production cost)
IoT_ar	Adoption rate of IoT technologies (speed of implementation)
IoT_sv	Saving on variable costs with IoT technology
IoT_iss_fq	Frequency of issues related to the adoption of IoT technology
IoT_iss_mn	Mean of issues (% of turnover) related to the adoption of IoT technology
IoT_iss_ex	Exponent of issue (power law) related to the adoption of IoT technology

Table 2. Simulation's outcomes

Output	Meaning
mean_IOT_adop	Mean level of adoption of IoT technologies
mean_IoT_incl_fail	Mean level of adoption of IoT technologies in failed firms
mean_IoT_incl_survived	Mean level of adoption of IoT technologies in surviving firms
mean_q	Mean quality strategy of survived firms
mean_mu	Mean price strategy of surviving firms
pct_firms_survived	Share of initial firms that survived
num_IoT_issues	Number of IoT issues
mean_IoT_Issues	Mean impact of IoT issues

The simulation results are analyzed using Python 3.8, on an Anaconda environment.

4. RESULTS

This section divides as follows. Firstly, the effect of the benefit of IoT on firms' strategies is shown. Secondly, the relationship between the IoT risk and the rate of survival of firms at the end of the simulation is investigated. The last part illustrates the effect of the parameters on the difference in risk preferences related to the adoption of IoT technologies between surviving and not surviving firms.

4.1. Firms' Strategies

This section investigates some notable relationships between environmental conditions and firm strategies that appeared in the simulation dataset. These results are computed by observing the behavior of the mean of a dependent variable on a dependent variable, employing a simple 2D plot.

Figure 7 and Figure 8 respectively show the relationship between the mean price and quality strategy of the firms that survived in the model, and the way it was affected by the effect of the introduction of IoT technology on the variable cost reduction.

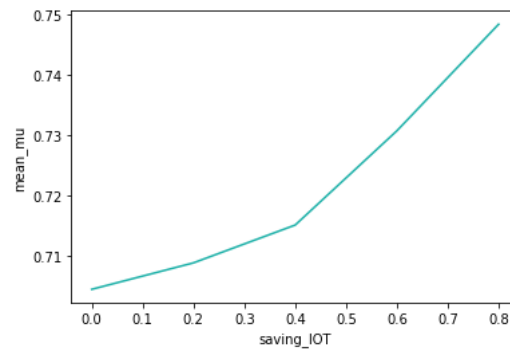


Figure 7. Firm's mean price and quality strategy

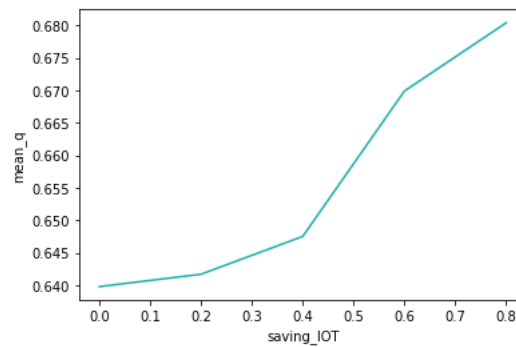


Figure 8. Firm's mean price and quality strategy in presence of IoT

Figures show that it exists a growing relationship between the quality and price strategies (axis of ordinates) and the variable costs reduction related to the IoT (abscissa axis). At a first sight, this result seemed contradictory. Since a firm should hypothetically focus on price or quality, we expected at least one of the relationships to be inversed. Nevertheless, we found two potential explanations for this phenomenon. First, the higher was the effect of IoT on the costs, the more likely were for firms to survive if they adopted a price strategy, that could be more competitive (because of the increase in the margins). Second, the savings in the variable costs affected the price adopted by the firm, because of the markup strategy. It implied that the higher was the savings due to IoT, the more profitable would be a firm on the market, and consequently the more profitable was a quality strategy. Since the results in Figure 8 were computed only on firms that survived at the end of the simulation, it became logical that the positive effect of IoT improved the fitness of a quality strategy in that specific competition environment.

4.2. Iot Risk and Firm Survival

This section investigates the effect of IoT inclination on the probability of survival of firms. Specifically, we investigated the mean IoT inclinations of firms that survive and the connection between the number of IoT issues and the share of firms that survived until the end related to the risk connected to the IoT. Logically, the higher the risk of IoT attack, the higher the risk of implementation of IoT technologies. We observed that this is a non-linear relationship. More precisely, the number of issues related to IoT technologies increases with the share of firms surviving, but they decrease after reaching a peak. Figure 9 displays this connection.

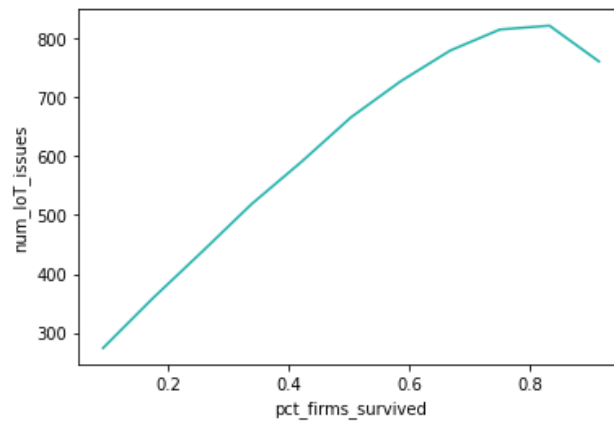


Figure 9. Shares of surviving firms and IoT issues occurring

We explain this phenomenon as the result of two concurring strengths. On the one side, the more firms survived, the more firms can implement IoT technologies. Consequently, the number of IoT issues increases. Nevertheless, after a given threshold, the market context is such that also allows firms that do not implement IoT technologies to survive. Consequently, the total number of issues in the simulation is below the peak value.

4.3. Difference in IoT Adoption Inclinations between Surviving and not Surviving Firms

In this paragraph, we investigate the difference between the mean IoT adoption inclination of survived firms and the mean IoT adoption inclination of failed firms. This value was computed for each resulting simulation, and for simplicity, we refer to it as “*difference*”. The values of the difference in the sampling were distributed as it is shown in Figure 10.

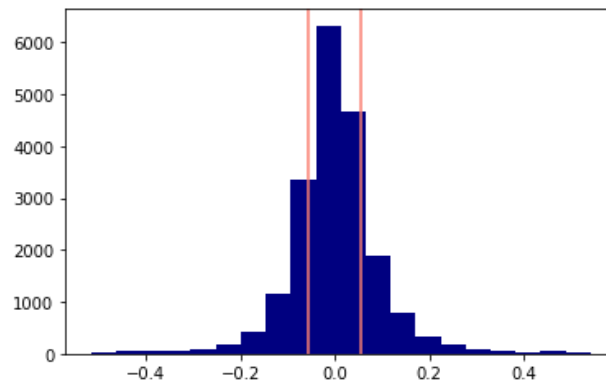


Figure 10. “difference” distribution at the end of simulations

We identified two notable groups of simulations. We called “*over*” the samplings in which the value of the difference is above the 80th percentile, and “*under*” the samplings in which the value of the difference is above the 20th percentile. In figure 10, these are represented by the two vertical lines. In this way, we could isolate two groups in which the difference of inclination affected the rate of survival, both positive and negative.

Figure 11 provides an overview of the effect of IoT adoption inclination on the overall success of firms.

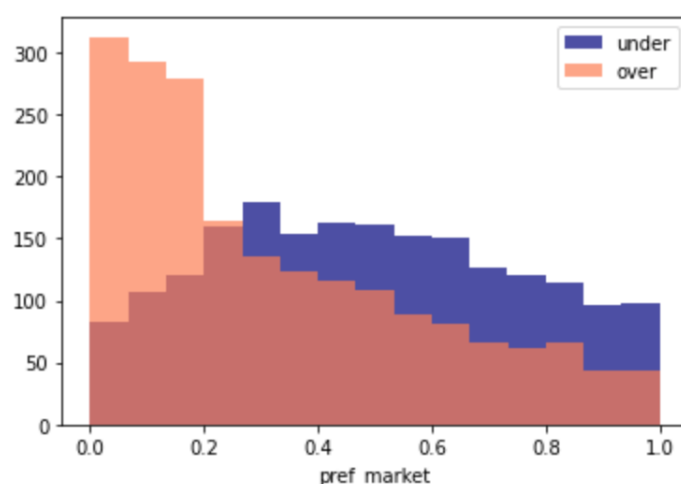


Figure 11 IoT adoption and success rate of firms

Figure 11 displays the distribution of market preferences for simulations in which firms have high and low *difference*. It is observable that the distributions had different shapes and different moments. More precisely, the *under* sampling had a mounded market preference distribution, while the *over* sampling had a j-shaped distribution, which resembled a power law distribution. The overlapping of the distributions showed that when the market strongly preferred the quality over the price, the competition process selected more likely firms with higher preferences in adopting IoT technologies.

We interpreted this phenomenon as the consequence of the lower production costs connected to the adoption of IoT technologies gives higher market shares to firms that invested in IoT. This market share grew progressively with the decrease of the parameter *market preferences* (when the relevance of price for costumers' decision increased). Therefore, the profit increased, and with it the probability of not failing.

5. DISCUSSION AND CONCLUSIONS

If we look at the results of the model in search of possible practical implications, the first result to consider is the positive influence that the introduction of IoT brings both to applications of cost leadership strategies and to those focused more on increasing the quality of the offer (figure 7 and 8). This allows us to consider as positive and desirable the effects of a high diffusion of IoT in enterprises, and suggests that it is desirable to search for initiatives and policies that favour its diffusion.

A second aspect that should not be underestimated is the particular success that the IoT seems to have in markets characterised by price competition. A success that manifests itself in increase of competitiveness in firms adopting the IoT than others. In these contexts, support for the introduction of IoT technologies could therefore be a precise strategy for both economic and political entities interested in supporting and guaranteeing greater success of their productive sectors in contexts of strong competition on production costs and sales prices that are common in the context of international globalisation where many companies operate today.

However, there are also elements to pay attention to. Figure 9 shows that in a context of wider firms survival and wider spread of IoT, the number of attacks and cybercrimes increases.

Although this does not seem to lead to a reduction in the number of surviving enterprises, it could lead to a reduction in market profit. It would therefore be necessary to look for methods and solutions to mitigate the occurrence of these cyber-attacks.

6. FUTURE DEVELOPMENTS

The model proposed in this paper can be used as a basis for various future implementations. A first hypothesized extension is the possibility of obtaining a more precise estimation of risk preferences by companies through the use of a dedicated questionnaire that would allow for detailed profiling of real cases and actors. Starting from this data, it would be possible to obtain a greater definition of the results previously shown, with the possibility of defining them within specific economic sectors or territorial areas. It would also then be possible to analyse how the actual adoption of the IoT by businesses could vary as the risk associated with the introduction of the IoT changes.

Furthermore, it could be possible to implement a similar model introducing a “criminal” agent, which could or could not invest in IoT technology, and then assess if there are some interesting co-adaptation dynamics in the risk preferences of firms. Finally, it could be possible to investigate if the results of this model would change with different conditions. For example, if an increase of uncertainty given by demand with a given trend or seasonality would increase or decrease the resulting risk preferences of firms related to the adoption of IoT.

Further simulations related to the possibility of simulating the results of policies favorable to the introduction of the IoT in competitive markets could also be included, to verify whether they could be used as a tool for promoting or protecting an economic system by both economic and political entities.

In the future, it would be desirable to hypothesize possible impact analyses of specific IoT architectures and projects dedicated to specific economic sectors modelled based on real market parameters in order to realistically simulate the benefits that can be hypothesized from the proposed innovations and to allow the related opportunities to be fully assessed.

REFERENCES

- [1] Edquist, H., Goodridge, P. and Haskel, J., 2019. The Internet of Things and economic growth in a panel of countries. *Economics of Innovation and New Technology*, 30(3), pp.262-283.
- [2] Al-Sarawi, Shadi, et al. "Internet of Things market analysis forecasts, 2020–2030." 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). IEEE, 2020.
- [3] Siozios, K., Anagnostos, D., Soudris, D. and Kosmatopoulos, E. IoT for Smart Grids.
- [4] Giang, N. K., Lea, R., Blackstock, M., & Leung, V. C. (2016, December). On building smart city IoT applications: a coordination-based perspective. In *Proceedings of the 2nd International Workshop on Smart* (pp. 1-6).
- [5] Govindraj, V., Sathiyarayanan, M., & Abubakar, B. (2017, August). Customary homes to smart homes using Internet of Things (IoT) and mobile application. In *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)* (pp. 1059-1063). IEEE.
- [6] Song, Y., Yu, F. R., Zhou, L., Yang, X., & He, Z. (2020). Applications of the Internet of things (IoT) in smart logistics: A comprehensive survey. *IEEE Internet of Things Journal*.
- [7] Porru, S., Misso, F. E., Pani, F. E., & Repetto, C. (2020). Smart mobility and public transport: Opportunities and challenges in rural and urban areas. *Journal of traffic and transportation engineering (English edition)*, 7(1), 88-97.

- [8] Furdik, K., Lukac, G., Sabol, T., & Kostelnik, P. (2013). The network architecture designed for an adaptable IoT-based smart office solution. *International Journal of Computer Networks and Communications Security*, 1(6), 216-224.
- [9] Dian, F. J., Vahidnia, R., & Rahmati, A. (2020). Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A Survey. *IEEE Access*, 8, 69200-69211.
- [10] Peraković, D., Periša, M., & Cvitić, I. (2015). Analysis of the IoT impact on volume of DDoS attacks. XXXIII Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju–PosTel, 2015, 295-304.
- [11] Singh, S., & Singh, N. (2015, October). Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 1577-1581). Ieee.
- [12] Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal*, 5(2), 1184-1195.
- [13] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3), 7280-7287.
- [14] Osservatori.net. 2021. Buon compleanno Internet (of Things). [online] Available at: <<https://www.osservatori.net/it/eventi/on-demand/convegni/buon-compleanno-internet-of-things>>.
- [15] Arnold, C., & Voigt, K. I. (2019). Determinants of industrial internet of things adoption in German manufacturing companies. *International Journal of Innovation and Technology Management*, 16(06), 1950038.
- [16] Luthra, S., Garg, D., Mangla, S. K., & Berwal, Y. P. S. (2018). Analyzing challenges to Internet of Things (IoT) adoption and diffusion: An Indian context. *Procedia Computer Science*, 125, 733-739.
- [17] Carcary, M., Maccani, G., Doherty, E., & Conway, G. (2018, September). Exploring the determinants of IoT adoption: Findings from a systematic literature review. In *International Conference on Business Informatics Research* (pp. 113-125). Springer, Cham.
- [18] Bilgeri, D., & Wortmann, F. (2017). Barriers to IoT business model innovation.
- [19] Maçik, R. (2017). The adoption of the internet of things by young consumers—an empirical investigation. *Economic and Environmental Studies*, 17(2 (42)), 363-388.
- [20] Tu, M. (2018). An exploratory study of Internet of Things (IoT) adoption intention in logistics and supply chain management: A mixed research approach. *The International Journal of Logistics Management*.
- [21] Chalapathi, G. S. S., Chamola, V., Vaish, A., & Buyya, R. (2021). Industrial internet of things (iiot) applications of edge and fog computing: A review and future directions. *Fog/Edge Computing For Security, Privacy, and Applications*, 293-325.
- [22] Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, 173, 114598.
- [23] Lager, A., Papadopoulos, A., & Nolte, T. (2020, September). IoT and Fog Analytics for Industrial Robot Applications. In 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (Vol. 1, pp. 1297-1300). IEEE.
- [24] Yudiansyah, A., Keke, Y., & Veronica, V. (2020). CAN THE MOBILE ROBOT BE A FUTURE ORDER-PICKING SOLUTION?: A CASE STUDY AT AMAZON FULFILLMENT CENTER. *Advances in Transportation and Logistics Research*, 3, 800-806.
- [25] Sengupta, J., Ruj, S., & Bit, S. D. (2020). A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. *Journal of Network and Computer Applications*, 149, 102481.
- [26] Deogirikar, J., & Vidhate, A. (2017, February). Security attacks in IoT: A survey. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 32-37). IEEE.
- [27] Atlam, H. F., & Wills, G. B. (2020). IoT security, privacy, safety and ethics. In *Digital twin technologies and smart cities* (pp. 123-149). Springer, Cham.
- [28] Minoli, D., & Occhiogrosso, B. (2018). Blockchain mechanisms for IoT security. *Internet of Things*, 1, 1-13.
- [29] Kshetri, N. (2017). Can blockchain strengthen the internet of things?. *IT professional*, 19(4), 68-72.
- [30] Bao, Z., Shi, W., He, D., & Chood, K. K. R. (2018). IoTChain: A three-tier blockchain-based IoT security architecture. *arXiv preprint arXiv:1806.02008*.

- [31] Alphand, O., Amoretti, M., Claeys, T., Dall'Asta, S., Duda, A., Ferrari, G., ... & Zanichelli, F. (2018, April). IoT Chain: A blockchain security architecture for the Internet of Things. In 2018 IEEE wireless communications and networking conference (WCNC) (pp. 1-6). IEEE.

AUTHORS

Francesco Bertolotti

I'm a industrial engineer and PhD candidate at LIUC university, currently visiting at the TU/e. My field of expertise is the simulation of risk preferences adaptation in socio-economical systems, especially employing agent-based modelling. I am member of the Complex System Society.



Riccardo Occa

I'm a industrial engineer and PhD candidate at LIUC university. My field of expertise is Logistic and Supply Chain 4.0, with a particular focus on IoT and Blockchain technologies.



BLOCKCHAIN ENABLED DIABETIC PATIENTS' DATA SHARING AND REAL TIME MONITORING

Dodo Khan, Low Tan Jung, Manzoor Ahmed Hashmani
and Moke Kwai Cheong

Department of Computer and Information Science, Universiti Teknologi
PETRONAS (UTP), Seri Iskandar 32610, Malaysia

ABSTRACT

According to the World Health Organization worldwide diabetes report, the number of diabetic patients has surged from 108 million in the 1980s to 422 million in 2014. According to researchers, the numbers will continue to climb in the next decades. Diabetes is a sickness that requires long-term self-care and close monitoring to be appropriately put under control. As a result, continuous monitoring of blood sugar levels has the potential to save millions of lives. This paper proposes a Blockchain-based platform that connects the patients, healthcare practitioners (HP), and caregivers for a continuous monitoring and care of diabetic patients. It lets the patients to securely connected to HP for the purpose of remote patient monitoring (telemedicine), whilst preserving patient data privacy using the blockchain technology. IoT sensors are used to read sugar levels and store these data in a tamper-proof immutable ledger (Hyperledger). This platform provides an End-to-End movement of the patient's data. That is, from the point where it is formed (sensors) to the point it ends up in the HP side. It gives patient a control-and-track function to maintain/track data movement. It provides a unique feature in allowing the patient to keep track of the private data and to pick who they want to share the data with and for how long (and for what reason). The platform is developed in two stages. Initially, the concept is implemented using the Hyperledger Fabric. Then, a Blockchain based on a novel Proof-of-Review (PoR) consensus model is included on to provide efficient performance and scalability in the Hyperledger fabric. Essentially, this proposed platform is to alleviate the pain points in traditional healthcare systems in the scopes of information exchange, data security, and privacy maintenance for real-time diabetic patient monitoring.

KEYWORDS

Blockchain, Consensus Protocols, Secure Data Movement, Real-time Monitoring, Hyperledger.

1. INTRODUCTION

Blockchain technology is one of the most talked-about breakthroughs in decentralised network with a bright future. It draws attention since the Nakamoto Bitcoin concept in 2008 [1]. Blockchain will account for 10% of global GDP by 2027 [2], according to the World Economic Forum [3]. Blockchain [4]. introduces a revolutionary data storing, monitoring, and transaction (digital transaction) procedure between two parties that does not require the use of a third-party broker. Aside from cryptocurrencies, Blockchain has gained momentum in a variety of areas with significant advancement including insurance [5], healthcare [6-8], IoT [4, 9, 10], supply chain, transportation, and certain government agencies. Blockchain with IoT has compounded the healthcare industry with an exponential growth. Indeed, the implementation of Blockchain has

enhanced the healthcare industry through numerous applications such as remote patient monitoring, real-time patient data aggregations or processing, medication supervision, and so on. Among others, remote patient monitoring is gaining popularity due to the COVID-19 pandemic situation that patients are not advisable or not safe to be admitted to hospitals for treatment.

It is predicted that chronic disease-related deaths would increase by 17 percent over the next ten years, affecting about 64 million individuals, and diabetes is regarded as one of the serious chronic disorders. The number of diabetic patients has increased from 108 million in the 1980s to 422 million in 2014, according to the World Health Organization's global diabetes report [11]. Researchers predict that the number will continue to rise in the coming decades. According to the International Diabetes Federation (IDF), diabetes prevalence among persons aged 20 to 79 was 9.3% in 2019, with a predicted increase to 10.9 percent, or approximately 700 million people, by 2045 [12] [13]. Diabetes is a serious public health issue in Malaysia. Diabetes is expected to affect 21.6 percent of Malaysia's adult population by 2020 [14] [15]. Diabetes, if left untreated, can lead to a number of major health consequences, including heart disease, renal failure, nerve and blood vessel damage [16]. Diabetes is a severe condition that requires long-term self-care and close monitoring for it to be properly controlled. The absence or lack of continuous monitoring, in terms of regular testing of blood glucose levels and being checked at the appropriate frequency by HP, the patient's life could be in jeopardy. Continuous monitoring of diabetes and blood sugar level has the advantage in saving lives.

It has been acknowledged that the use of ICT in healthcare data exchange increases the volume of data that floods the Internet. This is causing additional concerns in data security and data privacy. We believe that the patient should be allowed to regulate his or her health data because these data can be very sensitive and can be crucial to a patient per se. Examples existed in documents, testifying unauthorized disclosures and leakages of personal healthcare data. According to the Health Insurance Portability and Accountability Act (HIPAA), 13,236,569 medical records were compromised in 2018, more than double the 5,138,179 records exposed in 2017 [17]. In the Proetus Breach Barometer study [18], 140 million medical data were compromised in 2015.

Decentralization, anonymity, tamper resistance, auditability, transparency security, immutability, and trust less infrastructure are all properties of blockchain. These features may be used on the Internet of Things (IoT) as a critical technology [19] to overcome healthcare data exchange with tempered-proof interoperability for the benefit of the healthcare business operations.

For permissioned/private settings, blockchain technologies such as Ethereum [20, 21], Hyperledger Fabric [22], and Corda are available. The Hyperledger Fabric [23] [24] is the most well-known and commonly utilised platform for commercial applications. It has been thoroughly evaluated in a variety of business settings, including supply chain management, healthcare [25], and so on. Over 400 proof-of-concept and commercial distributed ledger applications in a variety of sectors and use cases have been implemented [24, 26]. HLF (Hyperledger Fabric) is a private (permissioned) blockchain system with an architecture that may be used to build industry-based blockchain applications [24] [27]. The HLF is adaptable, allowing for the addition or removal of components as needed, such as consensus and membership services. It uses the Docker container approach to allow smart contracts (chaincode) to create the application logic for the system [28]. The network's transactions are kept secret by using a channel isolation approach, which ensures that only authorised nodes of a certain channel may see the transaction.

This paper proposes a Blockchain-based platform that brings together stakeholders such as patients, HP, and caregivers for the continuous monitoring of diabetic patients. It lets patients to securely connect to HP for remote patient monitoring (telemedicine) and preserving patient data privacy with blockchain technology. Essentially, the purpose of the proposed platform is to

alleviate the pain points in traditional healthcare systems in the scopes of information exchange, data security, and privacy maintenance leveraging on real-time diabetic patient monitoring. The platform allows IoT sensors to sense sugar levels and store the sensed data in a tamper-proof immutable ledger (Hyperledger). This paper focuses on the End-to-End movement of the patient's data from the point where it is formed (sensors) to the point that ends up in the healthcare provider's side. The patient is granted a control-and-track feature to maintain and track his/her data movement. This feature is a unique approach that allows the patients to keep track of their private data and to decide who they want to share the data and for how long (and for what reason). This concept will be implemented using Hyperledger Fabric, and then, a novel Proof of Review (PoR) consensus model [29] Blockchain will be built to improve the efficiency, performance, and scalability of the Hyperledger fabric.

The rest of the paper is organised as follows. Sections 2 discuss the related work, section 3 and 4 discuss the proposed model and implementation of proposed model, and section 5 conclude the paper.

2. RELATED WORK

Some of the research works related to the proposed approach are briefly described in this section. The study in [30] proposed a 3-phase Blockchain enabled diabetes detection system that includes: registration, user identification using HER, and IoT data upload with Blockchain. After completing all three phases, machine learning algorithms were introduced to identify diabetes in patients and securely exchange the data within healthcare practitioners.

In [31], IoT, Blockchain, and cloud technologies were used to provide healthcare and tele-medical laboratory services in a hospital setting. IoT sensors collect and communicate vital signs and physiological information, allowing clinicians to give relevant, transparent, and safe medical treatment to their patients. This decentralised platform employs the Ethereum hybrid network certification approach, which provides a faster response time and lower cost than alternative methods. In a secure healthcare environment, communication is created between IoT nodes, servers, and the blockchain network. A front-end web application allows users to connect to the blockchain network.

According to the researchers in [32], BlockIoT is a solution that employs blockchain technology to communicate previously unavailable and centralised data from medical equipment to EHR systems, providing clinicians with better insight and improving patient outcomes. The Application Programming Interface (API) includes a customisable endpoint for all incoming medical device data, a distributed file system for data resilience, and knowledge templates for analysing, identifying, and displaying medical device data to providers.

The research in [12] looked examined diabetes patients' use of smartphones, as well as their plans to use them for self-care, monitoring, and management. The majority of participants in the study have a mobile phone (97.5%) and a smartphone (87%) and use the Internet on a daily basis (83.5 percent). The majority of participants utilised apps for meal planning (85.5%), glucose monitoring (76.5%), and scheduling diabetes appointments (76.5%). (76.5 percent).

A blockchain-based infrastructure proposed in [33] used an off-chain storage option to facilitate medical data exchange. Only critical information will be stored on the blockchain network, whilst all medical are stored on the cloud. The cloud operation module oversees all activities involving with the cloud storage. The recording module has six indications, including blood sugar, medication, nutrition, weight, exercise, and sleep, as well as indicator recording capabilities.

According to the experts in [34], there are a variety of ways that IoT and Blockchain technologies may be employed in the healthcare industry to improve overall performance and improve the present sector. Three major areas (healthcare) where their IoT and Blockchain technology might be used are (a) remote patient monitoring, (b) drug traceability, and (c) medical records management. The revolutionary usage of IoT and blockchain technologies in the healthcare business was also looked into.

HealthMudra [35] is an algorithm for developing the diabetes prevention guidelines. Health Mudra comprises a blockchain-based platform with optimization and machine learning algorithms. Diabetes can be avoided by following physicians' advice and reducing the symptoms. A decentralised Blockchain database was used to store information obtained from a large number of doctors to ease diabetes symptoms. Patients who use blockchain take ownership of their medical records.

The work in [36] presented a solution for addressing inefficiencies in existing techniques for exchanging healthcare data by employing a data-sharing system called MedChain, which mixes blockchain, digest chains, and a structured P2P network. MedChain was utilised to create a session-based healthcare data-sharing strategy that allows for data sharing flexibility. The results of the evaluation suggested that MedChain can improve productivity while also meeting data security needs.

A framework presented in [16] showed the integration of IoT and Blockchain to collect health data and share the data with healthcare organisations for daily smart therapy. To guard against hostile devices, the entire team concentrated on patient privacy and device security. Consequently, the patient may be guaranteed that his health data is collected on a regular basis, and in certain cases, automatically.

The Internet of Things (IoT) is being hailed as a game-changer in the healthcare industry, and the project in article [11] was to assess and analyse how IoT technology and its solutions might help patients with chronic conditions live better lives. The findings suggested that IoT can help in continuous glucose monitoring, as well as tracking patients' activities and diet to improve their lifestyle.

Blood sugar measurements may be gathered from distant CGMs and retrieved remotely utilising an IoT CGM-based system for mHealth as reported in the works of [37]. As a result, this technology allows patient monitoring and warning in the event of a potentially dangerous situation. Thanks to the blockchain and the recommended CGM-based system, it is feasible to supply a transparent and trustworthy blood sugar data source from a population in a quick, flexible, scalable, and low-cost manner. New mHealth apps for diagnosis, patient monitoring, and even public health activities might be enabled by crowdsourced data, all of which could help advance diabetes control and raise worldwide awareness of the disease's expanding prevalence.

A considerable majority of noncommunicable disease (NCD) diagnoses are erroneous, undesirable, or unnecessary, according to a study by researchers in [38]. As a result, they suggested a Proof of Disease (PoD) consensus technique based on Ethereum that includes a single instance of truth that computers can comprehend. It addresses a number of issues that have yet to be addressed by electronic health records (EHR) and health information exchange (HIE). This medical system will assist in meeting all of P6 medicine's complicated requirements (participatory, personalised, proactive, preventative, predictive, and precision medicine) and there by lessen sickness load.

Utilising the blockchain-based smart contracts, the works in [39] suggested a unique platform for patients' vital signs monitoring. The system was designed and built using Hyperledger fabric for enterprise-distributed ledger platform applications. This technique provides patients with a comprehensive, immutable medical history record, as well as a global access to medical information at any time and from any locations. The Libelium e-Health toolset was used to collect physiological data. A common benchmark tool called Hyperledger Calliper was utilised to assess the performance of the intended (and developed) system.

SMEAD system [40], an end-to-end safe solution for aiding diabetes patients. It contains devices that measure a number of markers, enabling for the tracking and forecasting of a patient's diabetes status. In the recommended approach, a MEDIBOX was used to configure the correct dosage and send an alert to consumers reminding them to take their prescription on time. The insulin dosage is stored at a safe temperature and is tested on a regular basis using the process described above. A Blockchain-based disruptive technology that offers cryptographic security and formalised data access through smart contracts for medical communities was carefully built to keep all data secret and to allow access to this data by physicians and other trustworthy parties. An alert is sent to caregivers via social media in the event of an emergency, such as skipping a medication, having abnormal blood sugar levels, or any security breach.

Table 1: Illustrates the critical comments on similar platforms and comparison with the proposed system.

Ref	Year	Critical Comment
[30]	2021	Blockchain enabled diabetes detection system to consist of 3-Steps. After completing all steps, machine learning algorithms were used to identify diabetes in patients.
[31]	2020	IoT sensors collect and communicate vital signs and physiological information, allowing clinicians to give relevant, transparent, and safe medical treatment to their patients. This decentralised platform employs the Ethereum hybrid network certification approach and A front-end web application allows users to connect to the blockchain network.
[32]	2021	BlockIoT is a solution that employs blockchain technology to communicate previously unavailable and centralised data from medical equipment to EHR systems, providing clinicians with better insight and improving patient outcomes.
[12]	2021	Here examined use of smartphones by diabetic patient. The findings are majority of participants utilised apps for meal planning (85.5%), glucose monitoring (76.5%), and scheduling diabetes appointments (76.5%).
[33]	2021	There are 6 six indications, including blood sugar, medication, nutrition, weight, exercise, and sleep stored on off0chain storage option to facilitate medical data exchange. Only critical information will be stored on the blockchain network, whilst all medical are stored on the cloud.
[35]	2020	HealthMudra is an algorithm for developing the diabetes prevention guidelines. A decentralised Blockchain database was used to store information obtained from a large number of doctors to ease diabetes symptoms. Patients who use blockchain take ownership of their medical records. It also utilises ML algorithm,
[36]	2019	MedChain was utilised to create a session-based healthcare data-sharing strategy that allows for data sharing flexibility. The results of the evaluation suggested that MedChain can improve productivity while also meeting data security needs.
[16]	2018	A framework in this showed the integration of IoT and Blockchain to

		collect health data and share the data with healthcare organisations for daily smart therapy including Blood sugar levels.
[11]	2019	The project was to analyse IoT technology, and its solutions might help patients with chronic conditions live better lives. The findings suggested that IoT can help in continuous glucose monitoring, as well as tracking patients' activities and diet to improve their lifestyle.
[37]	2018	New mHealth apps for diagnosis, patient monitoring, and even public health activities might be enabled by crowdsourced data, all of which could help advance diabetes control and raise worldwide awareness of the disease's expanding prevalence. The blockchain and the recommended CGM-based system is feasible to supply a transparent and trustworthy blood sugar.

3. PROPOSED MODEL ARCHITECTURE

This paper proposes a healthcare platform by utilizing the IoT devices (sensors) and the Blockchain technology. The platform comprises of 4 different layers. IoT (Sensor) devices layer, Gateway layer, Blockchain, and the Application layer. IoT devices layer comprises of sensors to measure blood sugar and other necessary vitals sign for diabetic patient, and it enables users (patients) to communicate the data to next layer. Gateway layer configures IoT devices (sensors) and connects them to the next layer. Basically, it allows IoT devices to pass data to the Blockchain layer. Moreover, Blockchain layer comprises of Blockchain related services which include, consensus mechanism [41], user identity management, IoT sensors information, distributed ledger storage, and smart contracts. The distributed ledger is a shared and replicated ledger, that is distributed across the entire Blockchain network. Every participant stores a copy of the ledger. Any changes in the distributed ledger should be updated in entire Blockchain after achieving consensus on the state of the ledger in within the ledger. This distributed ledger stores the patients' vital sign data collected by the IoT sensors. The smart contract is a program/code triggered by itself or external input to manage, store, access and modify the data on the distributed ledger. There may be many smart contracts, depending on the requirements of the diabetic patients i.e., event or notification, it will be triggered on certain time to notify patients to measure data or sending notification to HP every time a new Block is added to the ledger. API provides the designing services in the healthcare blockchain platform, allowing clients to interact with the app and control the blockchain network. The application layer is a user interface for visualizing vitalsign data for managing and controlling of healthcare devices. Basically, the Blockchain technology enables all the stakeholders to communicate, and share required data in a secure way. A P2P network, consensus methods, and asymmetric cyphers are used to communicate in the blockchain. The last part of Blockchain layer is Remote patient monitoring (RPM). It allows patients and HPs to connect with each other in the event of any emergency or unbalance data.

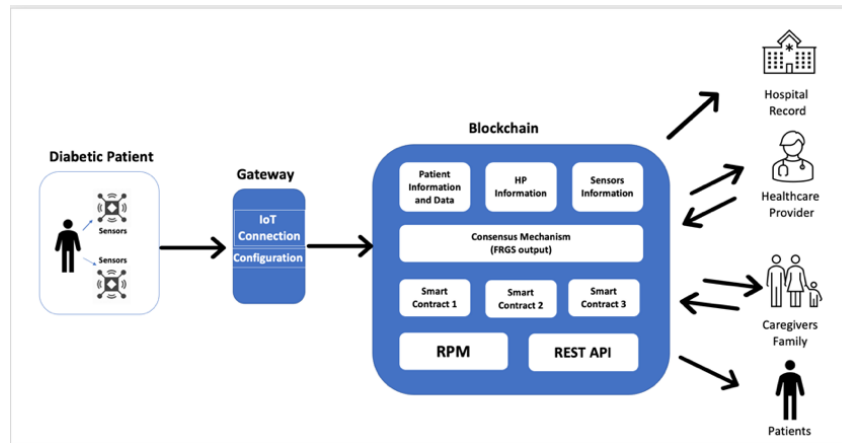


Figure 1: The proposed conceptual architecture

4. IMPLEMENTATION OF PROPOSED MODEL

The implementation of the proposed model is divided into two different phases. In the 1st phase, implementation will be done using the infamous private blockchain tool – Hyperledger fabric, and in 2nd phase, a custom Blockchain will be developed and designed using our novel Proof of Review consensus model. The performance, efficiency and scalability will be compared in both phases. Figure 2 illustrates the implementation of proposed Blockchain enabled diabetic patient monitoring platform in the Hyperledger Fabric. In this scenario, the Hyperledger Fabric is used as Blockchain platform along with healthcare sensors to measure vital signs. The essential components in Figure 2 are described below.

IoT Devices: IoT devices are different healthcare sensors, which may include blood pressure sensor, SPO2 sensors, glucometer sensor, body temperature sensor etc. As per the literature, all diabetes vital signs are effects or may get effected by blood glucose. Therefore, it is essential to monitor all these important vital signs. Here, individual (patient) will use IoT devices to measure all the vital data and forward them to Hyperledger (Blockchain).

Gateway: Raspberry Pi (RPi) defines as a series of single-board computers that are to connect the IoT devices. In this project Raspberry Pi will be used at the Gateway working as a bridge between Hyperledger and the IoT devices. Basically, it creates connection and configuration with sensors and Blockchain network.

Hyperledger Fabric: All required tools and components are deployed using the HLF LTS version. This comprises organisations with six peers: two committing peers (4 committing peer) and two endorsing peers (2 endorsing peer), as well as four CouchDB instances. With the ordering service, the RAFT consensus process would be used. Hyperledger provides the Representational State Transfer (REST) Application Programming Interface (API) capabilities to expose services to client applications for additional analysis. The client application, notification, may access all of the services written in the smart contract using REST API. In addition, the fabric client communicates with the fabric network via Google Remote Procedure Calls (gRPC). For various functions, several smart contracts will be established.

Smart Contracts: The business logic for smart contracts in transactions is defined by the Hyperledger Fabric. The Hyperledger Fabric Client SDK Node.js was used to connect with Hyperledger Fabric, and the smart contracts in this study were written in GO.

- SC1: It will create different notifications i.e., reminder, notifying HP to view data, notification on any unbalance data.
- SC2: It will allow patient to track their data and give permission/access to HP for certain predefined time.
- SC3: It will allow HP to write feedback and input on any data.

Application: All stakeholders, including patients and healthcare providers, will have access to data from the application side. This layer will either be a web application or a mobile application. This will provide aggregated data from the patients' vital signs. This layer also allows users to connect to healthcare services via the telemedicine component.

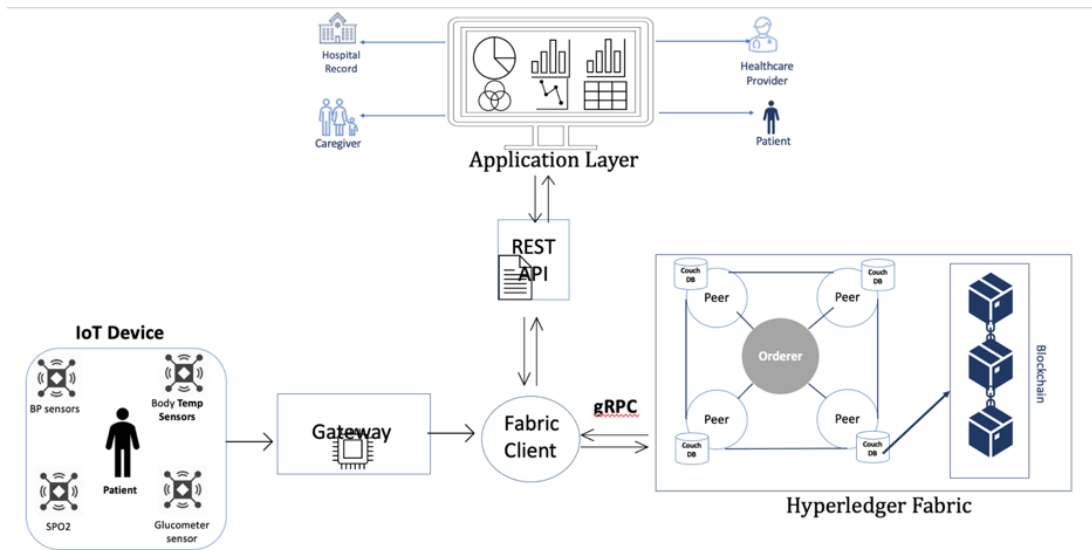


Figure 2: Implementation model of Blockchain enabled diabetic patients monitoring.

5. CONCLUSIONS

To conclude, this paper proposes a conceptual Blockchain-based framework that brings together stakeholders such as patients, HP, and caregivers for the continuous monitoring and care of diabetic patients. It comprises remote (diabetic) patient monitoring (telemedicine) features by utilizing blockchain technology. The IoT sensors will extract sugar levels, and possible other vital data, and store them in a tamper-proof immutable ledger (Hyperledger) focusing on the End-to-End movement of the patient's data from the point where it is formed (sensors) to the point where it ends up in the healthcare provider's side. The system gives the patient a control-and-track feature to maintain track of his/her data travel. It provides a unique feature that allows the patient to keep track of their private data and pick who they want to share it with and for how long (and for what reason). The implementation will take place in two phases. The initial concept will be implemented using Hyperledger Fabric, and then, a novel PoR Blockchain consensus model will be built to compare the efficiency, performance, and scalability with Hyperledger fabric. It should be mentioned here that in this paper only the first phase of implementation has been presented. Rest of the implementation will be presented in future works.

6. FUTURE WORK

The proposed approach is now undergoing implementation and in-depth/detailed studies. To evaluate and verify the suggested approach, experiments and simulations will be employed. The suggested approaches, the implementation, and experimental results are to be published in future articles.

ACKNOWLEDGEMENTS

The authors are grateful to Universiti Teknologi PETRONAS for providing the resources and materials needed to complete this research.

REFERENCES

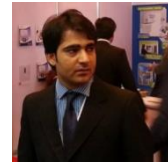
- [1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.
- [2] D. Khan, L. T. Jung, and M. A. Hashmani, "Systematic Literature Review of Challenges in Blockchain Scalability," *Applied Sciences*, vol. 11, no. 20, p. 9372, 2021.
- [3] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440-16455, 2020.
- [4] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, 2017: IEEE, pp. 618-623.
- [5] Z. Hess, Y. Malahov, and J. Pettersson, "Æternity blockchain," *Online*. Available: <https://aeternity.com/aeternity-blockchainwhitepaper.pdf>, 2017.
- [6] A. Ekblaw, A. Azaria, J. D. Halamka, and A. Lippman, "A Case Study for Blockchain in Healthcare: "MedRec" prototype for electronic health records and medical research data," in *Proceedings of IEEE open & big data conference*, 2016, vol. 13, p. 13.
- [7] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd international conference on open and big data (OBD)*, 2016: IEEE, pp. 25-30.
- [8] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of medical systems*, vol. 40, no. 10, pp. 1-8, 2016.
- [9] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the internet of things," *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 983-994, 2017.
- [10] J. Sun, J. Yan, and K. Z. Zhang, "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," *Financial Innovation*, vol. 2, no. 1, pp. 1-9, 2016.
- [11] A. M. Longva and M. Haddara, "How can iot improve the life-quality of diabetes patients?," in *MATEC Web of Conferences*, 2019, vol. 292: EDP Sciences, p. 03016.
- [12] A. Mehbodniya, A. Suresh Kumar, K. P. Rane, K. K. Bhatia, and B. K. Singh, "Smartphone-Based mHealth and Internet of Things for Diabetes Control and Self-Management," *Journal of Healthcare Engineering*, vol. 2021, 2021.
- [13] D. Atlas, "International diabetes federation," *IDF Diabetes Atlas, 7th edn. Brussels, Belgium: International Diabetes Federation*, 2015.
- [14] S. Bukhari *et al.*, "Transforming community-based screening of total hemoglobin using non-invasive devise," in *2016 IEEE Conference on Technologies for Sustainability (SusTech)*, 2016: IEEE, pp. 180-183.
- [15] A. M. Radzi, N. Draman, S. Yusoff, and R. Muhamad, "Depression and potential risk factors among the elderly with Type 2 Diabetes Mellitus in Kedah, Malaysia," *Medical Journal of Malaysia*, vol. 74, no. 2, pp. 103-108, 2019.
- [16] K. Azbeg, O. Ouchetto, S. J. Andaloussi, L. Fetjah, and A. Sekkaki, "Blockchain and IoT for security and privacy: A platform for diabetes self-management," in *2018 4th international conference on cloud computing technologies and applications (Cloudtech)*, 2018: IEEE, pp. 1-5.
- [17] H. D. B. Statistics, "Accessed: Jun. 11, 2019," ed.

- [18] M. Zhang and Y. Ji, "Blockchain for healthcare records: A data perspective," *PeerJ Preprints*, vol. 6, p. e26942v1, 2018.
- [19] K. C. Moke, T. J. Low, and D. Khan, "IoT Blockchain Data Veracity with Data Loss Tolerance," *Applied Sciences*, vol. 11, no. 21, p. 9978, 2021.
- [20] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2018: IEEE, pp. 264-276.
- [21] V. Buterin, "Ethereum white paper," *GitHub repository*, vol. 1, pp. 22-23, 2013.
- [22] E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1-15.
- [23] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, 2018: IEEE, pp. 51-58.
- [24] D. Khan, L. T. Jung, M. A. Hashmani, and M. K. Cheong, "Empirical Performance Analysis of Hyperledger LTS for Small and Medium Enterprises," *Sensors*, vol. 22, no. 3, p. 915, 2022.
- [25] S. Razdan and S. Sharma, "Internet of Medical Things (IoMT): Overview, Emerging Technologies, and Case Studies," *IETE Technical Review*, pp. 1-14, 2021.
- [26] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, "Performance modeling of hyperledger fabric (permissioned blockchain network)," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, 2018: IEEE, pp. 1-8.
- [27] S. Navroop, S. Nathalie, G. Alexandra, S. Robert, and G. Arianna, "Blockchain for Business-An Introduction to Hyperledger Technologies," *Tillgänglig online: [https://courses.edx.org/courses/course-v1:LinuxFoundationX+LFS171x+3T2017/course/\[Hämtad 2018-04-06\]](https://courses.edx.org/courses/course-v1:LinuxFoundationX+LFS171x+3T2017/course/[Hämtad%202018-04-06])*, 2018.
- [28] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017: IEEE, pp. 1-6.
- [29] D. Khan, L. T. Jung, and M. A. Hashmani, "Proof-of-Review: A Review based Consensus Protocol for Blockchain Application."
- [30] M. Chen *et al.*, "Blockchain-Enabled healthcare system for detection of diabetes," *Journal of Information Security and Applications*, vol. 58, p. 102771, 2021.
- [31] H. Wang, "IoT based clinical sensor data management and transfer using blockchain technology," *Journal of ISMAC*, vol. 2, no. 03, pp. 154-159, 2020.
- [32] M. Shukla, J. Lin, and O. Seneviratne, "BlockIoT: Blockchain-based Health Data Integration using IoT Devices," *arXiv preprint arXiv:2110.10123*, 2021.
- [33] Y. Liu, Z. Yu, and H. Sun, "Treatment Effect of Type 2 Diabetes Patients in Outpatient Department Based on Blockchain Electronic Mobile Medical App," *Journal of Healthcare Engineering*, vol. 2021, 2021.
- [34] P. Ratta, A. Kaur, S. Sharma, M. Shabaz, and G. Dhiman, "Application of blockchain and internet of things in healthcare and medical sector: applications, challenges, and future perspectives," *Journal of Food Quality*, vol. 2021, 2021.
- [35] R. Bhardwaj and D. Datta, "Development of a Recommender System HealthMudra Using Blockchain for Prevention of Diabetes," *Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries*, pp. 313-327, 2020.
- [36] B. Shen, J. Guo, and Y. Yang, "MedChain: Efficient healthcare data sharing via blockchain," *Applied sciences*, vol. 9, no. 6, p. 1207, 2019.
- [37] T. M. Fernández-Caramés and P. Fraga-Lamas, "Design of a fog computing, blockchain and IoT-based continuous glucose monitoring system for crowdsourcing mHealth," in *Multidisciplinary Digital Publishing Institute Proceedings*, 2018, vol. 4, no. 1, p. 37.
- [38] A. K. Talukder, M. Chaitanya, D. Arnold, and K. Sakurai, "Proof of disease: A blockchain consensus protocol for accurate medical decisions and reducing the disease burden," in *2018 IEEE SmartWorld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 2018: IEEE, pp. 257-262.
- [39] F. Jamil, S. Ahmad, N. Iqbal, and D.-H. Kim, "Towards a remote monitoring of patient vital signs based on IoT-based blockchain integrity management platforms in smart hospitals," *Sensors*, vol. 20, no. 8, p. 2195, 2020.

- [40] M. Saravanan, R. Shubha, A. M. Marks, and V. Iyer, "SMEAD: A secured mobile enabled assisting device for diabetics monitoring," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2017: IEEE, pp. 1-6.
- [41] D. Khan, L. T. Jung, M. A. Hashmani, and A. Waqas, "A Critical Review of Blockchain Consensus Model," in *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2020: IEEE, pp. 1-6.

AUTHORS

Dodo Khan. Currently, pursuing PhD in the field of IT in Universiti Teknologi PETRONAS Malaysia with the major in Blockchain. Mr. Khan have 8 years of industrial experience in many domains in especially Healthcare IT, his experience includes software development, software project management and strategic planning.



Mr. Khan obtain his Bachelor and master's degree in Software Engineering from Pakistan.

Dr. Tang Jung, Low obtained his bachelor's degree in Computer Technology from Teesside University (UK 1989).

MSc IT from National University Malaysia (2001), PhD IT from University Technology Petronas (Malaysia 2012). Low has been in the academic for more 25 years teaches various engineering and ICT courses. He is currently an Assoc. Prof. in UTP Computer and Information Sciences Department. research interest includes wireless sensor network, embedded systems, and IoT. Some of his current R&D include blockchain consensus model and IoT applications.



MANZOOR AHMED HASHMANI received the Ph.D. degree in communication networks from the Nara Institute of Science and Technology. He is currently with the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS. His research interests include computer communications (networks), artificial neural networks, and artificial intelligence.



SSL/TLS ENCRYPTED TRAFFIC APPLICATION LAYER PROTOCOL AND SERVICE CLASSIFICATION

Kunhao Li, Bo Lang, Hongyu Liu and Shaojie Chen

State Key Laboratory of Software Development Environment, Beijing, China

ABSTRACT

Network traffic protocols and service classification are the foundations of network quality of service (QoS) and security technologies, which have attracted increasing attention in recent years. At present, encryption technologies, such as SSL/TLS, are widely used in network transmission, so traditional traffic classification technologies cannot analyze encrypted packet payload. This paper first proposes a two-level application layer protocol classification model that combines packets and sessions information to address this problem. The first level extracts packet features, such as entropy and randomness of ciphertext, and then classifies the protocol. The second level regards the session as a unit and determines the final classification results by voting on the results of the first level. Many application layer protocols only correspond to one specific service, but HTTPS is used for many services. For the HTTPS service classification problem, we combine session features and packet features and establish a service identification model based on CNN-LSTM. We construct a dataset in a laboratory environment. The experimental results show that the proposed method achieves 99.679% and 96.27% accuracy in SSL/TLS application layer protocol classification and HTTPS service classification, respectively. Thus, the service classification model performs better than other existing methods.

KEYWORDS

SSL/TLS, HTTPS, Protocol Classification, Service Classification.

1. INTRODUCTION

SSL/TLS encryption technology has the advantages of high security and low cost and is widely used for secure communication of network applications. The protocol and service classification of encrypted network traffic are the basis of network service quality and network security technologies, which have received increasing attention. Since most of the content of the packets transmitted is encrypted, traditional traffic classification technology, such as deep packet inspection (DPI), has difficulty detecting SSL/TLS traffic^[1]. To solve the above problems, some researchers have focused on machine learning based methods. Because different applications and protocols have different functions, the statistical features of the generated traffic data are also different. Machine learning methods can find these differences and classify the traffic. Even though traffic is encrypted, its statistical features are still not affected, so the method can identify it.

In recent years, deep learning has made great achievements in computer vision and natural language processing. In the field of computer networks, technology has also attracted attention. Compared with traditional machine learning methods, this method does not require cumbersome

feature engineering. Instead, network traffic packets are directly inputted into the neural network, and the convolutional layers extract features to complete the classification task.

At present, research on application layer protocol classification of SSL/TLS encrypted traffic is still lacking. For service classification, traditional machine learning methods usually only extract features from the time and length of the network flow, while these methods do not make full use of the semantics of the packet content; the existing deep learning-based methods only use the first few packets of the SSL/TLS flow. The content is not portrayed from the global level of the flow. This paper comprehensively analyzes the characteristics of SSL/TLS single packet and session data and proposes a two-level application layer protocol classification model combining single packet and session. This model extracts features, such as entropy and randomness, from the ciphertext in a single packet and then classifies the protocol. According to the labels of packets in the same session, we build a voting model to determine the traffic protocol. For the problem of HTTPS service classification, we propose a method fusing the global session features and time sequence features, which fully utilizes the encrypted network flow information and improves the task's accuracy. The contributions of this paper mainly include the following:

1) We propose an SSL/TLS application layer protocol classification method combining ciphertext features and a voting mechanism. The method first extracts ciphertext features and uses a machine learning model to complete single-packet protocol determination. Then we use a voting scheme to realize the application layer protocol classification of SSL/TLS sessions.

2) We propose an HTTPS service classification method based on feature engineering and deep learning. This method establishes a CNN-LSTM model to extract the time-series features of the packets in the SSL session and merges them with the global features of the session.

3) In a laboratory environment, we construct the dataset from many sources, including Chrome, Foxmail, FileZilla, etc. We apply the two methods mentioned above to the dataset. The accuracy of the application layer protocol classification method achieves 99.679% and the accuracy of the HTTPS service classification method reaches 96.27%, which is better than the existing machine learning and deep learning methods.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 introduces the details of our proposed methods. Section 4 presents experimental results. Finally, the paper is concluded in Section 5.

2. RELATED WORK

In the early Internet, every application/protocol used a fixed port number assigned by the Internet Assigned Numbers Authority (IANA)^[2]. Therefore, according to the port field in the TCP/UDP header, the application types and protocol types of flow can be classified. For example, HTTPS uses port 443, and SMTPS uses port 456. In recent years, port-based methods have not been more effective than as previously were, because dynamic ports are widely used and new applications have emerged continuously. DPI classifies traffic through pattern matching on the payload in the packet, but it is still difficult to adapt to the encrypted network environment.

At present, research on the application layer protocol classification of network encrypted traffic is still lacking. Some network encryption traffic service classification methods have emerged, mainly including traditional machine learning-based and deep learning-based methods.

Traditional Machine Learning Methods: Because the statistical features of the traffic generated by different applications or services have certain differences in the spatial and temporal dimensions, machine learning methods can utilize the features to classify traffic. Such methods usually include two steps: feature extraction and model training. Features are mainly composed

of packet length features, packet ordering features, and packet timing features, which include the number of packet bytes, the packets' time interval, and the flow duration, etc. The models mainly include KNN, SVM and random forest, etc. These models work well on small datasets and do not rely on hardware. However, feature engineering requires much time and professional knowledge to support.

Lashkari et al.^[3] regarded unidirectional and bidirectional encrypted traffic flow as the units and extracted timing-related features such as flow duration and packet time interval to train KNN and C4.5 models, which classify different services of encrypted traffic. Dominik et al.^[4] used SVM to distinguish whether HTTPS traffic is a mail service. They extracted features, including the duration of the session, the different patterns of daily/weekly traffic usage, and the inherent periodicity.

Deep Learning Methods: This kind of method can automatically learn features and classify encrypted traffic. It does not rely on complex and high-cost feature engineering. The methods can directly deal with packet data and achieve good classification performance.

Wei et al.^[5] applied the end-to-end method to classify encrypted traffic for the first time. They proposed a one-dimensional CNN method. Lotfollah et al.^[6] first removed the ethernet header and conducted normalization of the packet. Then, they designed SAE and one-dimensional CNN models to classify the service type of traffic. Mingze et al.^[7] proposed a text-based convolutional neural network (Text-CNN). He et al.^[8] proposed an image-based convolutional neural network (Image-based CNN). They are also better than traditional machine learning methods in service classification.

In addition, RNNs and their variant models have also achieved satisfactory results in service classification. Zhuang et al.^[9] combined a CNN with a LSTM and extracted the packet features and the sequence features to classify the encrypted traffic service. Haipeng et al.^[10] proposed two models, an attention-based LSTM and a hierarchical attention network (HAN) to model sequential traffic. Liu et al.^[11] proposed attention-based bidirectional GRU networks to solve the problem of HTTPS traffic classification. The bidirectional GRU layer is used to extract the forward and backward features of the byte sequence in the session, and the attention layer assigns weights according to the contribution of features to the classification.

3. METHODOLOGY

There are still relatively few achievements in the current encryption traffic classification studies for application layer protocol identification. In addition, service classification methods cannot comprehensively describe the characteristics of network flows. Therefore, we propose a two-level traffic classification framework to solve these problems. For the application layer protocol classification task, we consider the characteristics of ciphertext and propose a classification method combining single packet features and session features; for the service classification task, we extract the global features and time sequence features of the flow, and propose a CNN and LSTM-based classification model, which describes the flow from a comprehensive perspective.

3.1. Framework

The SSL/TLS protocol consists of two layers (as shown in Table 1). The bottom layer is the SSL/TLS record protocol, which is responsible for encrypting packets with a symmetric key. The upper layer is the SSL/TLS handshake protocol, which is divided into four subprotocols:

Handshake Protocol, Change Cipher Spec Message Protocol, Alert Message Protocol and Application Data Protocol.

Table 1. Structure of SSL/TLS protocol

Record Layer		
Content Type	Version	Length
Handshake Protocol (Content Type= 0x16)		
Change Cipher Spec Message (Content Type = 0x14)		
Application Data (Content Type = 0x17)		
Alert Message (Content Type = 0x15)		

The SSL/TLS record protocol consists of content type, version and length fields. The content type field represents the subtype of the recording protocol. The version field represents the version of the SSL/TLS protocol. Content type and length fields represent the type and length of the remaining packet content, respectively. For instance, if Content Type=0x16, the rest content is the content of the handshake protocol; if Content Type=0x17, the rest is the encrypted data in the transmission phase.

Figure 1 shows the framework of the two-layer classification model proposed in this paper. We regard SSL/TLS sessions as the detection units. For application layer protocol classification, we trained the protocol classification model for the five most widely used encrypted protocols in the current network environment: HTTPS, FTPS, SMTPS, IMAPS, and POPS. For service classification, FTPS, SMTPS, IMAPS and POPS only carry a single service (FTPS is used for file transfer, and the other three protocols are used for mail). Therefore, these protocols can be directly output as service types. Only the HTTPS protocol carries multiple services(browser, streaming, etc.), so we focus on HTTPS service classification. Therefore, we propose a convolutional and recurrent neural network-based model combining global and sequential features (CRNN-CGSF) to realize HTTPS service classification.

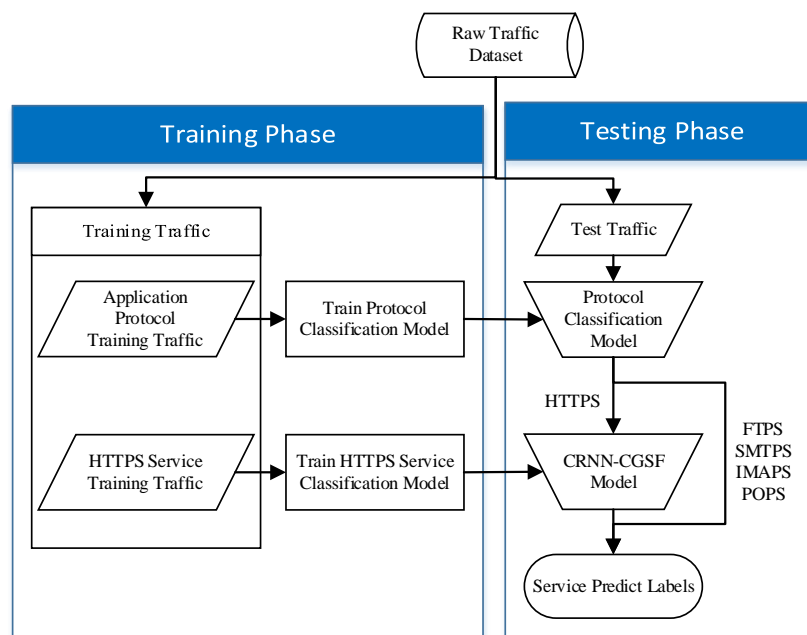


Figure 1. Framework of application protocols and services classification architecture

As shown in Figure 1, our method is divided into a training phase and a testing phase. In the training phase, the dataset is divided into an application layer protocol training dataset and an HTTPS service training dataset. The protocol classification model and the HTTPS service classification model (CRNN-CGSF) are independently trained. In the testing phase, the protocol classification model is used to identify the application layer protocol, and then the sessions classified as HTTPS are further used to classify the service using the CRNN-CGSF model. Finally, the services of all sessions are output.

Our detection unit is the session. Therefore, the raw flow first needs to be restored to a session before detection. We define the session based on a four-tuple <source IP, source port, destination IP, destination port> (because the protocols are all TCP, so there is no need to express the protocol) bidirectional flow.

According to the protocol specification, handshake phase packets and application data protocol packets should be in a complete session. The standard handshake phase should conform to <client_hello, server_hello, server_hello_done, client_key_exchange, change_cipher_spec> mode or <client_hello, server_hello, change_cipher_spec> mode. Incomplete SSL sessions usually do not have complete handshake phase information or data transmission due to being truncated or from network delays. In addition, we discard this type of flow.

3.2. Application Layer Protocol Classification

In this paper, the application layer protocols include HTTPS, FTPS, SMTPS, IMAPS, and POPS, and their plaintext protocols (HTTP, FTP, SMTP, IMAP, and POP) have different format specifications according to the RFC. Thus, we believe that the data will have different distributions in randomness and entropy after encryption to distinguish different application layer protocols.

The application layer protocol detection framework for SSL/TLS encrypted traffic is shown in Figure 2. The input of the framework is a preprocessed SSL session. Detection is mainly divided into three steps: feature extraction, single-packet classification, and voting:

- (1) **Feature extraction.** We extract all application data protocol packets in every session because these packets contain SSL/TLS header information and are the first packets of a single forward or backward flow in the encrypted data transmission phase. Therefore, the encrypted data of these packets provide the most sufficient format information of the corresponding plaintext protocol. Second, we extract the packets' application data field (i.e. encrypted data) and perform feature extraction on each encrypted data.
- (2) **Single packet classification.** The features extracted from each packet are input into the classifier. The classifier will output the application layer protocol label corresponding to each application data protocol packet in the session.
- (3) **Voting.** Since only one application layer protocol is used in the same session, we vote on the prediction result of a single packet and select the application layer protocol with the highest frequency as the application layer protocol used in the session.

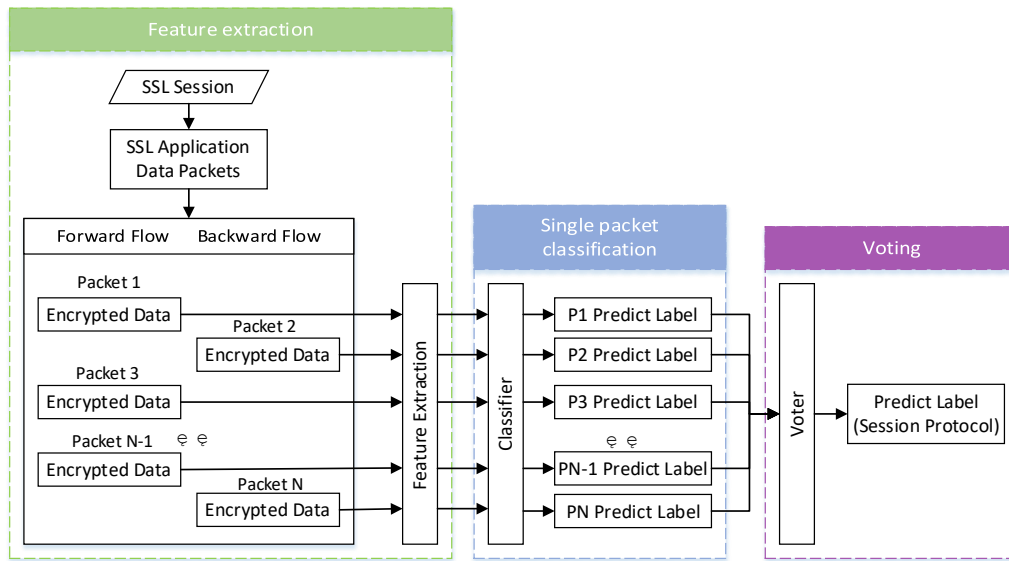


Figure 2. Framework of the application protocol detection

3.2.1. Feature Design

We refer to the randomness detection scheme defined in the National Institute of Standards and Technology (NIST) standards^[12] and the ciphertext entropy theory mentioned in the literature [13] and design the features including randomness measurement, entropy, length and bidirectional flow.

Randomness measurement is an important indicator for evaluating the randomness of ciphertext. When the plaintext content of different packet formats is encrypted, the distribution of these characteristic values is still different. Entropy can be used to indicate the uniformity of the ciphertext's byte distribution. The more uniform the byte distribution is, the higher the entropy is^[14]. We also use the length feature because different protocols have different length distributions. For example, the lengths of FTPS-Data packets and HTTPS packets are usually hundreds of bytes or even reach the MTU. The lengths of other protocol packets are relatively short. The bidirectional flow features are mainly for FTPS-Data packets, which are all unidirectional in the SSL/TLS encrypted transmission stage, while other protocols are usually bidirectional. The details of these features are shown in Table 2:

Table 2. List of features of the SSL/TLS application package

Feature Type	Feature	Description
Randomness Measurement	Frequency	To detect the proportion of zeroes and ones for the entire sequence.
	Frequency within a Block	To detect the proportion of ones within M-bit blocks.
	Runs	To detect the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits.
	Discrete Fourier Transform	To detect the peak heights in the Discrete Fourier Transform of the sequence.
	Non-overlapping Template Matching	To detect the number of occurrences of pre-specified target strings.

	Overlapping Template Matching	To detect the number of occurrences of pre-specified target strings.
	Linear Complexity	To detect the length of a linear feedback shift register (LFSR)
	Serial	To detect the frequency of all possible overlapping m-bit patterns across the entire sequence.
	Approximate Entropy	To detect the frequency of all possible overlapping m-bit patterns across the entire sequence.
	Cumulative Sums	To detect the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence.
	Random Excursions	To detect the number of cycles having exactly K visits in a cumulative sum random walk.
	Longest Run of Ones in a Block	To detect the longest run of ones within M-bit blocks.
Entropy	Byte entropy of inter-packet	To measure the randomness of the byte frequency between ciphertext binary packets.
Length	Packet Length	The length of the current packet.
	SSL packet length	The Content-Type field of SSL handshake packet's header.
Bidirectional	Bidirectional	Indicates if the encrypted data transmission is bidirectional or not.

3.2.2. Packet Classification Model

In the selection of classification models, we consider the currently popular machine learning algorithms, which are mainly divided into three categories:

- (1) **Traditional machine learning algorithm:** We use C4.5^[17], KNN^[18], LR and SVM^[19]. These algorithms have the advantages of fitting for nonlinear classification, supporting for numerical and discrete data, and preventing overfitting, and are widely used.
- (2) **Integrated learning algorithm:** We use Random Forest^[20], Vote, Adaboost, GBDT and XGBoost. Multiple weaker learners integrate these algorithms. Compared with single learners, they usually reach higher accuracy. Moreover, the robustness and generalization ability of these models have also been improved.
- (3) **Neural network algorithm:** MLP and DNN are used in this paper. MLP is an artificial neural network with a forward structure. It overcomes the weakness that a single-layer perceptron cannot recognize linear inseparable data. DNN is an improvement over MLP, and overcomes the problem of gradient disappearance caused by the increase of the number of network layers in the multilayer perceptron. In addition, it has more types of activation functions.

3.3. Service Classification

Among the existing service classification methods, machine learning methods only use the time and length features of the network flow. Deep learning methods only focus on the content of the first few packets of the session. Both of them lack a macro description of the entire network flow. Therefore, we propose the CRNN-CGSF model that integrates the global features of the session with the packet time sequence information in the session to solve HTTPS service classification. The model architecture is shown in Figure 3:

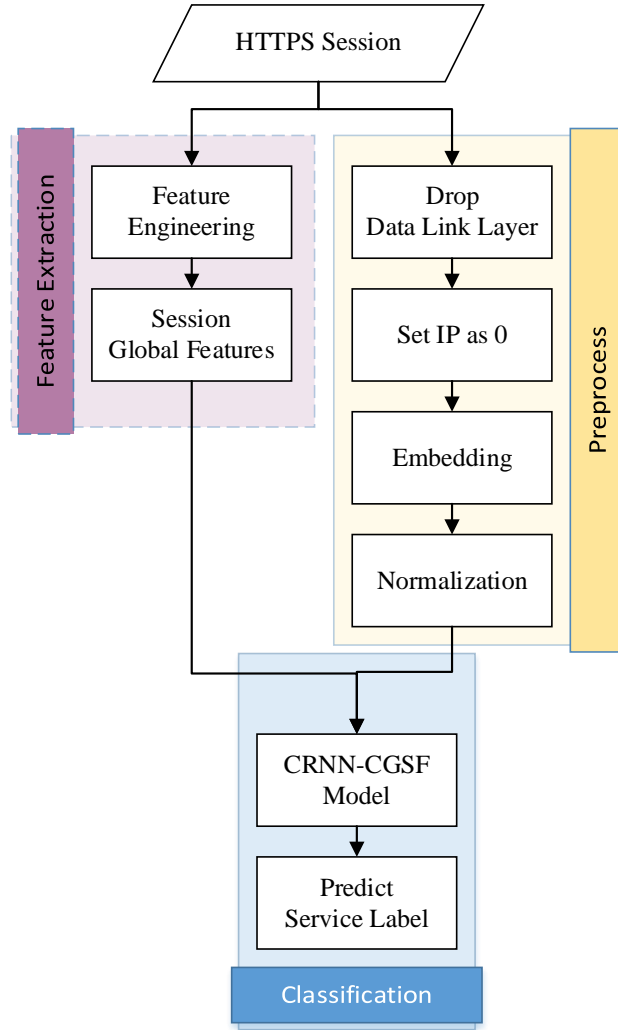


Figure 3. Framework of HTTPS service detection

We first conduct preprocessing and then perform feature extraction and regularization of HTTPS sessions. As shown in Figure 3, the left branch extracts the global features of the session, and the right branch preprocesses the packets in the session.

- (1) **Global feature extraction.** We refer to [3] to extract time-related features, including the duration of the flow (duration), forward interarrival time (fiat), backward interarrival time (biat), flow interarrival time (flowiat), active time (active), idle time (idle), flow bytes per second (fb_psec) and flow packets per second (fp_psec), totaling 23 dimensions.

- (2) **Packet preprocessing.** The packet's data link layer (Ethernet frame) contains the MAC address and the IP version. The MAC address is the host identifier and is useless for the task of network traffic classification, although it may affect the classification results; we only pay attention to the ipv4 version of the network traffic, so the IP protocol version is also useless. Thus, we discard the Ethernet frame. The source IP and destination IP in the network layer are unnecessary information, so we replace these fields of the IP header with zeroes. To reduce the input dimension of the model, we convert the bits in the data packet into bytes. Then we conduct normalization, that is, we divide all byte values by 255 and map them to the $[0,1]$ interval; the purpose of this is to obtain a better computing performance.
- (3) **Service classification model.** We input the extracted global features of the session and the preprocessed packets into the classification model. The model outputs the predicted HTTPS service label. The model structure is shown in Figure 4:

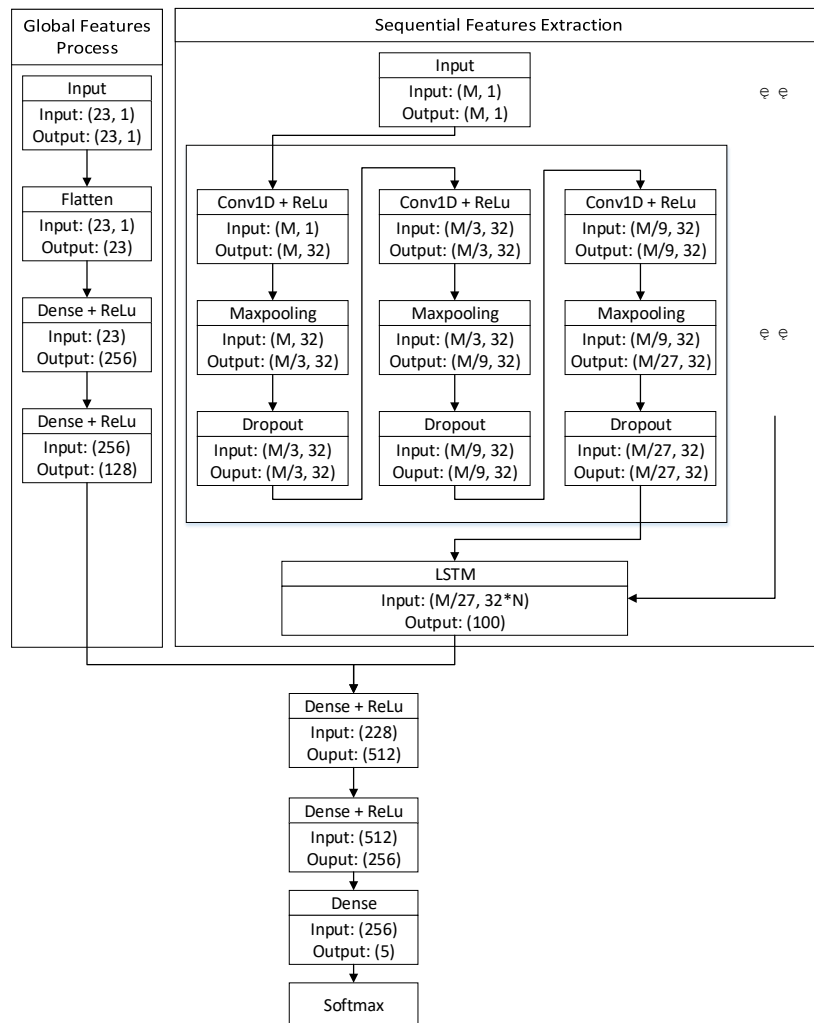


Figure 4. Architecture of the CRNN-CGSF model

In Figure 4, the left branch of the model processes the global features of the session, whose input dimensionality is 23×1 . First, the input is flattened to 23 dimensions. Then through two FC layers, the output is 128-dimensions vector. The multiple branches on the right extract the time sequence features from packets. The input is the first M byte stream of the first N packets of the session. The first N packets are selected because the first few packets of the session are in the handshake phase and contain certain semantic information. When the encrypted communication

is turned on, the semantics will be greatly weakened. We use CNN to extract the features of each packet byte stream. The CNN layer includes three concatenated Conv1D, Maxpooling, and Dropout. After that, the features of these sequential packets are concatenated and input into LSTM to extract the time sequence features. Then, 100-dimensional time sequence features and 128-dimensional global feature vectors are integrated and input to three FC layers. Finally, the classification result is output through Softmax.

4. EXPERIMENTS

4.1. Dataset

At present, most of the encrypted traffic classification datasets are used for widened meaning protocol classification (SSL, SSH, Tor, etc.). Our task is to subdivide the traffic of SSL/TLS, so we collect the corresponding network traffic as a dataset in a laboratory environment. We use Wireshark to capture the traffic, and the captured packets are stored in pcap.

We browse a large number of web pages with Chrome and Firefox to collect HTTPS traffic. For FTPS traffic, we use two personal computers, using one as an FTP server (encrypted by SSL) and the other as an FTP client. We use FileZilla on the client and remote access the FTP server for file upload and download operations. We enable SMTP/POP and SMTP/IMAP services in a personal QQ mailbox and a 163 mailbox and then use Foxmail to send, receive, and delete mails to collect SMTPS, POPS and IMAPS traffic.

We collect browser service traffic through Chrome and Firefox, but not all the traffic generated by the browser belongs to the browser service. For example, if we use NetEase Cloud Music, the traffic generated, by the browser, that transmits multimedia content belongs to the streaming service traffic. For other services, we use specific applications to collect the traffic data.

The total size of the dataset is 4.7GB and it contains 51 pcap files. The specific content of the dataset is shown in Table 3:

Table 3. List of captured protocols and applications

Service	Protocol	Content	Session Number
Browser	HTTPS	Chrome、Firefox	5908
Streaming	HTTPS	QQ Music、NetEase CloudMusic、Tencent Video	1596
Chat	HTTPS	Weibo Chat、Skype	1280
Mail	HTTPS	buaamail、Tom Mail	1070
	SMTPS	Foxmail	581
	POPS	Foxmail	961
	IMAPS	Foxmail	708
File Transfer	HTTPS	Skype、Baidu Netdisk、115 Pan	1068
	FTPS	FileZilla	378(FTPS-Control) 2088(FTPS-Data)

We split the dataset into two subdatasets according to two types of tasks: dataset-protocol and dataset-service. The former is used for the SSL/TLS application layer protocol classification task, and the latter is used for the HTTPS service classification task. The traffic of these two datasets is marked with application layer protocol labels and service labels in sessions.

In the upper application layer classification task, we divide FTPS into FTPS-Control and FTPS-Data. The reason is that the plaintext protocol formats of these two FTPS packets are completely

different, so there is a great difference in indicators such as entropy and randomness metrics. We can regard it as two subprotocols of FTPS.

4.2. Indicators and Experimental Settings

To evaluate the classification effects of different models, we use the accuracy rate (Acc) to evaluate the overall effect of multiclassification, using precision (Pr), recall (Rc) and F1 scores (F1) that comprehensively consider accuracy and recall to evaluate the effect of a certain type of classification in multiclassification.

We set up 4 groups of experiments:

- **Experiment 1:** We compare the classification effects of different machine learning algorithms on the SSL application layer protocol. As mentioned in Section 4.1, FTPS traffic can be divided into FTPS-Control and FTPS-Data. Therefore, according to FTPS as one class, or divided into two classes, we designed two schemes of five classes and six classes.
- **Experiment 2:** We compare our proposed CRNN-CGSF model with the existing service classification methods, and verify that our method has a higher accuracy.
- **Experiment 3:** We explore the impact of the input dimensions of the CRNN-CGSF model on the performance of HTTPS service classification. The input dimension is determined by the number of intercepted session packets N and the number of intercepted packet bytes M.
- **Experiment 4:** We explore the impact of introducing global features on the classification results of the HTTPS service.

4.3. Results and Analyses

4.3.1. Application layer protocol classification (Experiment 1)

Since the dataset is unbalanced in categories (as shown in Table 3), we sampled the dataset to train the model better. For the single-packet detection experiment, we select 8000 data from each protocol (for the case where the FTPS-DATA samples in the six categories are less than 8000, we select 1500 samples). Then, we divide them into a training set and a test set at a ratio of 4:1. We select 150 sets of session data from each protocol as the test set for the session detection experiment.

We use the machine learning algorithms mentioned in Section 3.2.2 to train the classifiers. The test results are shown in Table 4, 5, and 6:

Table 4. Accuracy of different traditional machine learning methods

		C4.5	KNN	LR	SVM
5 classes	Packet	0.74406	0.73496	0.43156	0.39971
	Session	0.98500	0.90866	0.69054	0.23491
6 classes	Packet	0.74496	0.72971	0.44264	0.41071
	Session	0.98772	0.92292	0.66243	0.33378

Table 5. Accuracy of different ensemble learning methods

		RF	VOTE	ADA	GBDT	XGBoost
5 classes	Packet	0.80659	0.81494	0.59589	0.79595	0.81828
	Session	0.99670	0.99679	0.94098	0.99688	0.99679
6 classes	Packet	0.80734	0.81518	0.56525	0.79764	0.81871
	Session	0.99554	0.99628	0.78296	0.99576	0.99650

Table 6. Accuracy of different neural network methods

		MLP	DNN
5 classes	Packet	0.55717	0.69721
	Session	0.78580	0.90317
6 classes	Packet	0.59157	0.70423
	Session	0.84665	0.91438

In traditional machine learning methods, C4.5 performs far better than other methods. In the six-classification task, the accuracy of single-packet detection and session detection reaches 0.74496 and 0.98772, respectively. In the ensemble learning methods, RF, VOTE, GBDT and XGBoost have similar accuracies in session detection tasks. XGBoost performs best in single-packet detection and six-classification session detection; GBDT performed best in five-classification session detection, with an accuracy that is 0.009% higher than XGBoost. In neural network methods, DNN is better than MLP because DNN has deeper layers and can better fit the data. Still, DNN has limited improvement performance and is far inferior to C4.5, KNN and integrated learning methods.

From the detection point of view, the accuracy of session detection is much higher than that of single-packet detection. Because the accuracy of single-packet detection reaches a certain height, the incorrect single-packet classification is corrected after voting. In terms of methods, integrated learning algorithms are generally better than traditional machine learning algorithms. Integrated learning can combine multiple single learners with a certain strategy, which greatly improves generalization performance. Among all the methods, XGBoost is the most comprehensive. The XGBoost single-packet detection and session detection results of each type of protocol are shown in Table 7, and the confusion matrix of the experimental results is shown in Figure 5:

Table 7. Detailed experimental results of XGBoost (5 classes)

	Packet			Session		
	Pr	Rc	F1	Pr	Rc	F1
FTPS	0.86496	0.79813	0.83020	0.99379	1.00000	0.99689
HTTPS	0.91086	0.89623	0.90349	0.99371	0.98750	0.99060
IMAPS	0.85053	0.75000	0.79711	1.00000	0.99375	0.99687
POPS	0.77748	0.89286	0.83118	0.99375	0.99375	0.99375
SMTPS	0.73533	0.78320	0.75851	0.99379	1.00000	0.99689

Actual Label	ftps	1281	8	71	66	195
	https	42	1450	20	26	53
	imaps	103	27	1165	167	129
	pops	23	48	55	1451	64
	smtps	53	86	56	180	1181
		ftps	https	imaps	pops	smtps
		Predict Label				

Figure 5. Confusion matrix of the single packet detection result of XGBoost

Observing the results of Table 7 and Figure 5, it can be seen that XGBoost has the best recognition effect on the HTTPS protocol in single-packet detection. The accuracy, recall and F1 score indicators reach 0.91086, 0.89623 and 0.90349, respectively; for the SMTPS protocol, the recognition effect is the worst, but its F1 score also reaches 0.75851. In terms of session detection, XGBoost has a very good classification effect for each protocol, and the F1 score can be stabilized above 0.99.

4.3.2. Comparative experiment on service classification methods (Experiment 2)

We compared the six methods in the five current papers^[3,4,5,13,14] containing the same kind of research with our own method. The experimental results are shown in Table 8:

Table 8. Comparative experimental results of HTTPS encrypted traffic service classification

Method	Acc
Our method(CRNN-CGSF)	0.9627
C4.5[3]	0.8905
KNN[3]	0.7030
1D-CNN[5]	0.9410
SAE[6]	0.9406
LSTM[15]	0.9080
nnDPI[16]	0.9401

According to Table 8, the accuracy of deep learning methods (1D-CNN^[5], SAE^[6], LSTM^[15] and nnDPI^[16]) is better than traditional machine learning methods (C4.5^[3] and KNN^[3]). Among these methods, our model has the best accuracy, reaching 0.9627, which is 0.0217 higher than the second-highest 1D-CNN.

4.3.3. CRNN-CGSF model input dimension experiment (Experiment 3)

To obtain better results in the classification of HTTPS services, we explored the influence of N (flow size) and M (intercept length) on the model classification effect. N represents the number of packets we select in the session. If the value of N is too small, the information in the handshake phase will be incomplete; if the value of N is too large, encrypted data will be used, which will have a certain negative impact on the model performance. M indicates how many bytes we choose from each packet. If the value of M is too small, then the information extraction

of each packet will be insufficient; if the value of M is too large, it will have a certain impact on the computational performance overhead of the model.

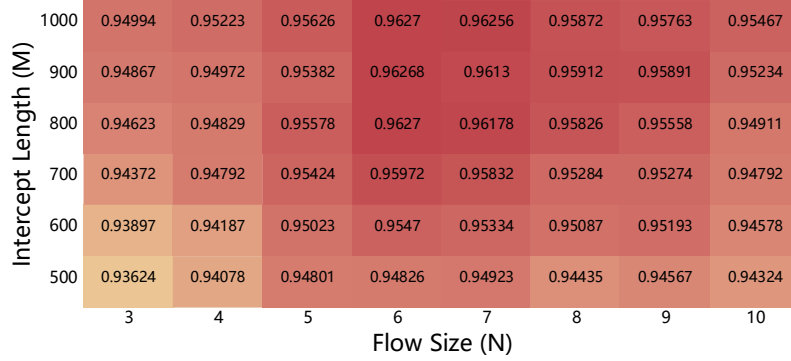


Figure 6. Thermodynamic diagram under different flow sizes and intercept lengths

Figure 6 shows that the accuracy of the model increases as the value of M increases. This is because the longer the intercepted byte length of a single packet is, the richer the information will be. The accuracy of the model first increases with increasing N and then decreases with further increasing N . $N=6$ can be regarded as a turning point because if the stream length is too long, then model will use the packets of encrypted data transmission, and accuracy will decrease. Because the packet carries ciphertext, it interferes with the model performance. The model achieves the maximum value at $N=6$, $M=800$ and $N=6$, $M=1000$. Considering the calculation performance of the model, we choose $N=6$ and $M=800$.

4.3.4. Validity experiment of introducing global features (Experiment 4)

To verify that the global features are effective, we performed a comparative experiment on global features. The model that does not contain global features is called CRNN-UOSF (convolutional and recurrent neural networks using only sequential features). Compared with the CRNN-CGSF, CRNN-UOSF eliminates the global feature input layer (Input), the flattened layer (Flatten) and the fully connected layer (Dense). The rest of the structure is the same. The experimental results are shown in Table 9:

Table 9. Comparison of experimental results between CRNN-CGSF and CRNN-UOSF

Method	Acc
CRNN-CGSF	0.9627
CRNN-UOSF	0.9504

The model's accuracy without global features is 0.9504, and the model's accuracy with global features is 1.23% higher than that without global features. Therefore, it can be concluded that the introduction of the global features of the session enables the model to better characterize the session.

5. CONCLUSIONS

This paper focuses on the application layer protocol classification and service classification of SSL/TLS encrypted traffic. We first extract features such as randomness and entropy of encrypted data for application layer protocol classification and then use a machine learning model

to judge single packets. After that, we utilize voting mechanisms to realize application layer protocol classification for SSL/TLS sessions. The experimental results show that XGBoost has the best comprehensive detection effect. We propose the CRNN-CGSF model combining session global features and packet time sequence features for HTTPS service classification. The model uses CNN and LSTM to effectively utilize the packet byte stream information. In addition, we improve the accuracy of the model by introducing the global features of the session. The experimental results show that the accuracy of our method can reach 96.27%, which is better than the existing traditional machine learning and deep learning methods. Our method can provide preliminary traffic analysis results in network management and QoS and can provide basic support for further analysis procedures. In future work, we will focus on the classification of new versions of protocols such as HTTP2 and QUIC. In addition, our experiments are based on the dataset collected in the laboratory, which may lead to the limitations of our model. We will pay more attention to the traffic in different network environments and further improve the generalization capabilities and robustness of our model.

REFERENCES

- [1] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, "A survey on encrypted traffic classification," in International Conference on Applications and Techniques in Information Security, (Berlin, Heidelberg), pp. 73–81, 2014.
- [2] Moore A W, Papagiannaki K. Toward the accurate identification of network applications[C]//International Workshop on Passive and Active Network Measurement. Springer, Berlin, Heidelberg, 2005: 41-54.
- [3] Lashkari A H, Draper-Gil G, Mamun M , et al. Characterization of Encrypted and VPN Traffic Using Time-Related Features[C]// The International Conference on Information Systems Security and Privacy (ICISSP). 2016.
- [4] Schatzmann D, Mühlbauer W, Spyropoulos T, et al. Digging into HTTPS: flow-based classification of webmail traffic[C]//Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. 2010: 322-327.
- [5] Wei W , Ming Z , Wang J , et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]// 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2017.
- [6] Lotfollahi M , Zade R S H , Siavoshani M J , et al. Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning[J]. Soft Computing, 2017.
- [7] Song M, Ran J, Li S. Encrypted traffic classification based on text convolution neural networks[C]//2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2019: 432-436.
- [8] He Y , Li W . Image-based Encrypted Traffic Classification with Convolution Neural Networks[C]// 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC). IEEE, 2020.
- [9] Zhuang Z , J Ge, Zheng H , et al. Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network[C]// 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2018.
- [10] Yao H, Liu C, Zhang P, et al. Identification of encrypted traffic through attention mechanism based long short term memory[J]. IEEE Transactions on Big Data, 2019.
- [11] Liu X , You J , Wu Y , et al. Attention-Based Bidirectional GRU Networks for Efficient HTTPS Traffic Classification[J]. Information encs, 2020, 541.
- [12] Rukhin A . A statistical test suite for random and pseudorandom number generators for cryptographic applications[J]. NIST Special Publication 800-22, 2000.
- [13] Mishra S, Bhattacharjya A . Pattern analysis of cipher text: A combined approach[C]// International Conference on Recent Trends in Information Technology. IEEE, 2013.
- [14] Wang Y, Zhang Z, Guo L, et al. Using entropy to classify traffic more deeply[C]//2011 IEEE Sixth International Conference on Networking, Architecture, and Storage. IEEE, 2011: 45-52.

- [15] Vu L , Thuy H V , Nguyen Q U , et al. Time Series Analysis for Encrypted Traffic Classification: A Deep Learning Approach[C]// 2018 18th International Symposium on Communications and Information Technologies (ISCIT). 2018.
- [16] Bahaa M , Aboulmagd A , Adel K , et al. nnDPI: A Novel Deep Packet Inspection Technique Using Word Embedding, Convolutional and Recurrent Neural Networks[C]// 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). 2020.
- [17] Quinlan J R. Improved use of continuous attributes in C4.5[J]. Journal of artificial intelligence research, 1996, 4: 77-90.
- [18] Abeywickrama T, Cheema M A, Taniar D. K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation[J]. arXiv preprint arXiv:1601.01549, 2016.
- [19] Nello Cristianini and John Shawe-Taylor. 1999. An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods. Cambridge University Press, New York, NY, USA.
- [20] Tin, Kam. The random subspace method for constructing decision forests.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1998, 20(8):832-832.

FROM MONOLITH TO MICROSERVICES: SOFTWARE ARCHITECTURE FOR AUTONOMOUS UAV INFRASTRUCTURE INSPECTION

Lea Matlekovic and Peter Schneider-Kamp

Department of Mathematics and Computer Science, University of Southern
Denmark, Odense, Denmark

ABSTRACT

Linear-infrastructure Mission Control (LiMiC) is an application for autonomous Unmanned Aerial Vehicle (UAV) infrastructure inspection mission planning developed in monolithic software architecture. The application calculates routes along the infrastructure based on the users' inputs, the number of UAVs participating in the mission, and UAVs' locations. LiMiC1.0 is the latest application version migrated from monolith to microservices, continuously integrated, and deployed using DevOps tools to facilitate future features development, enable better traffic management, and improve the route calculation processing time. Processing time was improved by refactoring the route calculation algorithm into services, scaling them in the Kubernetes cluster, and enabling asynchronous communication in between. In this paper, we discuss the differences between the monolith and microservice architecture to justify our decision for migration. We describe the methodology for the application's migration and implementation processes, technologies we use for continuous integration and deployment, and we present microservices improved performance results compared with the monolithic application.

KEYWORDS

autonomous UAV, mission planning, microservices, Docker, Kubernetes, CI/CD

1. INTRODUCTION

Infrastructure inspection is a dangerous, expensive, and time-consuming task. Falls from inspection sites are the leading cause of death among construction workers [1]. It is not the only risk construction workers face. They may also come into contact with toxic chemicals, moving machinery, speeding traffic, or high-voltage equipment. The risks increase when the infrastructure is in disrepair. Deploying UAVs to inspection eliminates the safety risks. Defects can be detected from the UAV camera and construction workers can see the state of an inspection site before they climb and start repairing the infrastructure. The UAV technology reduces the costs of inspection since it does not require special inspection equipment, helicopters nor airplanes [2]. Nowadays, many companies use UAVs for visual infrastructure inspection, but they still control them manually or use automated flying. By increasing the degree of UAVs' autonomy, we expect to reduce the costs even more and save time on process optimization. To develop an autonomous system for infrastructure inspection, we need to plan the inspection mission and schedule tasks for each UAV participating in the mission.

The preliminary system design for autonomous UAV infrastructure inspection was described in [3]. The system was designed in three layers: cloud services, UAVs, and communication between them. Global mission planning and scheduling software is deployed in the cloud. When the mission is calculated, the route coordinates are sent to the UAVs. Message exchange between UAVs and cloud services is through the HTTP (Hypertext Transfer Protocol). Robot Operating System (ROS) runs on UAVs where high-level control software is deployed. PX4 open-source software is used as a low-level flight controller.

Linear-infrastructure Mission Control (LiMiC) is presented in [4]. It is a software application for global mission planning and scheduling developed in a monolithic architecture. Monolithic means that all the application logic was composed into a single program. To facilitate further development and improve the route calculation processing time, we decided to redesign the architecture and compare the performance.

The paper is structured as follows. In Section 2 we compare monolithic architecture to microservices, justifying our choice for a redesign. We describe DevOps practices that facilitate microservices deployment to the production. Related work, concerning monolith decoupling is presented in Section 3. In Section 4 we describe LiMiC application logic, architecture redesign based on LiMiC functionalities, migration and implementation processes, technologies we used as well as the deployment strategy. In Section 5 we compare performance between the application developed in monolith and microservice architecture. We conclude in Section 6 discussing possible improvements and future work.

2. PRELIMINARIES

In this section, we describe software architecture styles relevant for this article and present their benefits as well as challenges. Software architecture in general describes application organization and structure. Architectural decisions impact application quality, performance, maintainability, and usability. Software architecture chosen at the beginning of the application development can have a high impact on the application's future and affect its success. Architecture should be considered and planned if the application needs to be scalable, maintainable, and easily upgradable. When developing the application from scratch, the focus is usually on having a working product as soon as possible. However, that approach can become unsustainable when the number of application features grows fast. If the architecture is not reconsidered, the development slows down and application becomes difficult to maintain.

2.1. Monolithic Architecture

Monolithic applications encompass several tightly coupled functions and tend to have a big codebase. The development does not require advanced architecture planning since the application is developed, packaged, and deployed as a standalone instance. The packaged application can be deployed to the server and scaled horizontally by running multiple instances behind a load balancer. The load balancer distributes the traffic across the instances deployed on the different servers. Testing the monolith applications can be performed end-to-end by launching the application and using existing testing framework e.g., Selenium [5]... However, when the code base grows and many developers work on the same application, monolith applications face challenges. The application can become too complex and difficult to understand. That prevents quick updates and development slows down. On each update, the entire application must redeploy and it can be difficult to track update impacts. It leads to extensive and time-consuming manual testing. A bug in any module can potentially bring the whole system down since all instances rely on

the same code base. Even though the application can be scaled, the load is usually not equally distributed to all the modules i.e., the traffic is directed to only one application module, but they are all scaled equally. When application modules have different resource requirements, like they often do, it can lead to unnecessary CPU (Central Processing Unit) and memory consumption. Monoliths are not robust to changes and adopting a new framework or technology would lead to rewriting of the entire code base which is both expensive and time-consuming [6]. Development in monolithic architecture could be a good choice if the application will not need to be further extended or as a starting point when the main goal is to have a simple, working end-product. However, when the development slows down as the codebase grows, it is recommended to reconsider the architecture [13].

2.2. Microservice Architecture

Microservice architecture, also known as Microservices, is a software architecture that structures an application as a collection of small, loosely coupled services. The services are independently deployable, highly maintainable, and testable [7]. The concept was developed to overcome the downsides of monolithic architecture. Microservices have clear boundaries between each other and communicate through the HTTP protocol, usually by exposing a REST API and sending requests. Each service represents one capability that makes it easier to understand and locate the code. It also makes them robust to changes. Since they are small and deployed independently, they are easy to update and maintain. Services can be scaled independently and automatically, depending on the load. They can use different technology stacks, including programming language and data storage. That gives high flexibility to the development teams. However, there are some drawbacks of microservice architecture. The fact that a microservices application is a distributed system requires handling of fallacies the distributed computing carries. It means that developers must deal with the additional complexity.

There are opinions suggesting to start application development in monolith architecture first and then migrate to microservices [8]. As monolithic systems become too large to deal with, many enterprises are drawn to breaking them down into microservices. On the other side, others recommend starting with microservices if that architectural style is the goal [9]. However, most large scale websites including Netflix, Amazon, and eBay have evolved from a monolithic architecture to microservices [10].

2.3. DevOps

Microservices, since they are independent, bring more complexity in application deployment than monolithic applications. There are many practices and tools developed to facilitate testing, integration, and deployment of microservices [11]. DevOps is a combination of practices and tools designed to facilitate the delivery of applications. It aims to increase an organization's ability to deploy applications faster by removing the barriers between development and operations teams [12]. DevOps practices automatize the delivery process and are implemented as a part of a production pipeline. Applied tools and practices depend on the application delivery requirements and goals. Continuous integration is the practice of merging changes to the main branch as often as possible. When developers commit local changes to the remote repository, automated build and tests can run there before proceeding to production. Remote repository platforms with built-in version control, like GitLab and GitHub, facilitate the collaboration between developers and enable continuous integration. These platforms also integrate with different DevOps tools to enable continuous delivery and deployment. The practice of continuous delivery

includes continuous integration and after a successful build, automatically propagates the application to staging. In staging, the application is deployed to the testing environment. There, the application including all services can be run and tested. Continuous delivery requires manual approval for release into production. However, the continuous deployment includes all the steps described in continuous integration and delivery, but instead of manual, the release process is also automated. Described process can vary in complexity depending on the number of services, test requirements, and in general, the deployment strategy.

3. RELATED WORK

The concept of microservices arose around ten years ago and today is a widely used concept for large enterprises to develop their software systems. According to IDC (International Data Corporation), by 2022, 90% of all new applications will be based on microservices architectures [13]. Adopting microservices improves agility and flexibility, enabling enterprises to bring their products and services to market faster.

Although the benefits of microservices are evident, adopting it is not an easy task as it usually involves refactoring the monolithic application. There are many critical questions to ask before deciding to refactor the monolith. Will the refactoring bring value? How to re-architect an existing system without having to stop all other work on it? How big should a microservice be? What are some of the migration patterns you could adopt when splitting up a monolith? [13] There is some research tackling these questions in the latest years. Most of the previous research on microservices either identifies challenges when splitting the monolith like in [14], or proposes refactoring methods. Classification of refactoring approaches is presented in [15]. There are four notable approaches identified. Static Code Analysis approaches require the application's source code analysis and derive a decomposition from it through possible intermediate stages. Meta-Data aided approaches require more abstract input data, like architectural descriptions in form of UML (Unified Modeling Language) diagrams, use cases, interfaces, or historical VCS (Version Control System) data. Workload-Data aided approaches aim to find suitable service cuts by measuring the application's operational data (e.g. communication, performance) on module or function level and use this data to determine a fitting decomposition and granularity. Dynamic Microservice Composition approaches try to solve the problem more holistically by describing a microservices runtime environment. Other than the previously mentioned categories, the resulting set of services is permanently changing in each iteration of recalculating the best-fitting composition (based on e.g. workload). Static and dynamic analysis of the monolith is further described and used for refactoring in [16]. The authors used dynamic software visualization to identify appropriate microservice boundaries for a real-world application. A Dataflow-driven approach is proposed in [17]. The article presents a semi-automatic mechanism to break business logic into microservices and visualize services with data flows. Although the refactoring algorithm could save some time in splitting the monolith, it still requires identification and description of monolith application business logic. All the research work we encountered based refactoring on application logic and we followed the same approach. Another interesting work is proposed in [18] where the focus is on profitability depending on the number of refactored services and deployment. The article shares lessons learned on an industrial migration to a web-oriented microservice architecture. The services are refactored based on capabilities and the decision on service size is based on the company organization and profitability. For deployment tools, Docker is emphasized as a widely used container technology for achieving service isolation. The article presents the return of investment from refactoring the

monolith and concludes by presenting the benefits of the transition for a given company.

Even though the previous research literature guided us in choosing the most suitable approaches, each application is different and requires an individual approach bringing new challenges when splitting it into microservices.

4. METHODOLOGY

In this section, we describe processes leading to LiMiC migration from monolith to microservice architecture. Migration from monolith to microservices is not an easy task [19]. There are many different approaches to splitting the monolith. The code should be studied before decoupling into the logical components. Dependencies should be identified and isolated. Even though our monolithic application was not deployed in the production and it did not have users depending on it, we encountered difficulties while refactoring, and parts of the code had to be rewritten instead of reused.

Main processes leading to LiMiC1.0 implementation from monolith to microservices are as follows:

- **Identification** - identification of features in monolithic application
- **Isolation** - isolation of features where each isolated feature is one meaningful and standalone unit
- **Implementation** - implementation of isolated features withing framework enabling the communication between them

For application deployment we developed a strategy and used DevOps tools described in this section to implement and automatize processes leading to deployment. By enabling continuous integration and continuous deployment, we tremendously benefited from the microservice architecture, i.e. better codebase organization, faster development, easier integration and deployment.

4.1. Identification & Isolation

Linear-infrastructure Mission Control (LiMiC) is an application software for autonomous Unmanned Aerial Vehicle (UAV) infrastructure inspection. It is a mission planner for calculating the near-optimal routes for power line inspection using UAVs. The user can choose the mission targets, i.e. power towers for inspection, calculate the order of visiting and generate waypoints. Waypoints represent the coordinates sent to the UAVs for navigation to the inspection targets.

We used data from the OpenStreetMap [20] to represent the power towers on the web interface and to build a graph using NetworkX Python package [21]. The web interface shows UAVs and towers on a 2D map. We used graph structure to describe and store path data. Nodes in the graph represent power towers. Nodes contain the tower's unique identifier and geographical location. Edges between the nodes contain distances between the neighboring towers which we used to describe the costs of flying from one tower to another. We used Google OR-Tools [22] to find the near-optimal visiting schedule for multiple vehicles visiting a set of inspection targets stored in the graph. When a user selects inspection targets, the vehicle scheduling algorithm generates a distance matrix using the A* algorithm. A* finds the shortest path, based on the distance data stored in the graph, for each combination of UAVs and towers participating in the mission and returns it to the vehicles scheduler. Calculated paths are used for determining the visiting schedule, i.e. which UAV will inspect which tower. The result is visualized on the map and paths are stored to be sent to the UAVs. Paths are represented as a set of waypoints containing geographical locations.

We developed the application in monolithic architecture structured as Python modules situated in a folder. The frontend communicates with the backend through HTTP requests using Flask Python framework [23]. As the application was growing and many developers started contributing, we started thinking about redesigning the structure. We downloaded the data from the OpenStreetMap and saved the graph locally. If we wanted to update the graph we would need to run the graph generation manually and save a new one. Since we plan to grow the application and deploy it to production, it was necessary to redesign the graph creation and update it automatically in defined time intervals. The codebase was not organized in a logical structure and it was difficult to navigate. Addition of a new feature required searching through the codebase to understand the processes before contributing to the code. A lot of time that could be spent on the development was invested in understanding the existing implementation. We wanted to facilitate the feature addition and reduce the time spent on understanding the code. That required a fixed structure, but flexible and adaptive to feature addition. In order to grow the application and deploy it to production, we decided to restructure it. Another concern was the load distribution. Some parts of the application are more static while others are heavily used. If deployed to the production, some parts of the application would have to be unnecessarily scaled. Therefore, we structured the application in microservices based on the application's capabilities. The proposed structure is shown in Figure 1. The arrows determine the communication flow between the services, e.g. the graph service sends a request to the towers service while the towers service provides power tower information needed to create a graph. The tower service should contain the capability to extract the data from the OpenStreetMap, organize it in a suitable form and save it to the database. The same capabilities should exist for other infrastructures like bridges and railways. Bridges can be reached by calculating the route near the power lines or railways leading to the bridge location. Depending on the infrastructure we want to inspect, we would build the graph containing nodes with locations of power towers, railway towers, or bridges as defined in the OpenStreetMap. The data is used by the web interface to visualize the inspection targets. Graph service would then use available data to create a graph. Inspection targets chosen on the web interface would be sent to the vehicle routing problem (VRP) solver service, which uses the A* algorithm to find the shortest path between the targets and UAVs and determines the routing schedule. A* calculates the shortest path based on the graph created in the graph service.

4.2. Implementation

In the previous subsection, we identified services logically, based on the capabilities of the monolithic application. In this subsection, we describe the process of monolith decoupling and implementation in detail.

The decoupling process was divided into two parts. First was the graph creation based on the data stored in a database created in railways, towers, and bridges services. That way, we were able to test the generated graph file with a monolithic application. The second part was splitting the routing solver and A* pathfinder into independent components. Every time the request comes to the routing solver, it requests the shortest path for each set of targets and UAV locations from the A*. A* requests a graph from the graph service only on the first request. All the services use FastAPI [24] web framework for exposing APIs (Application Programming Interfaces) to the web interface or other services. FastAPI enables input validation where we define how the input is structured. It facilitates the input usage in the code without installing additional packages. FastAPI provides interactive, automatically generated documentation which we find very useful for endpoint visualization and testing. When looking into the performance comparison, FastAPI is

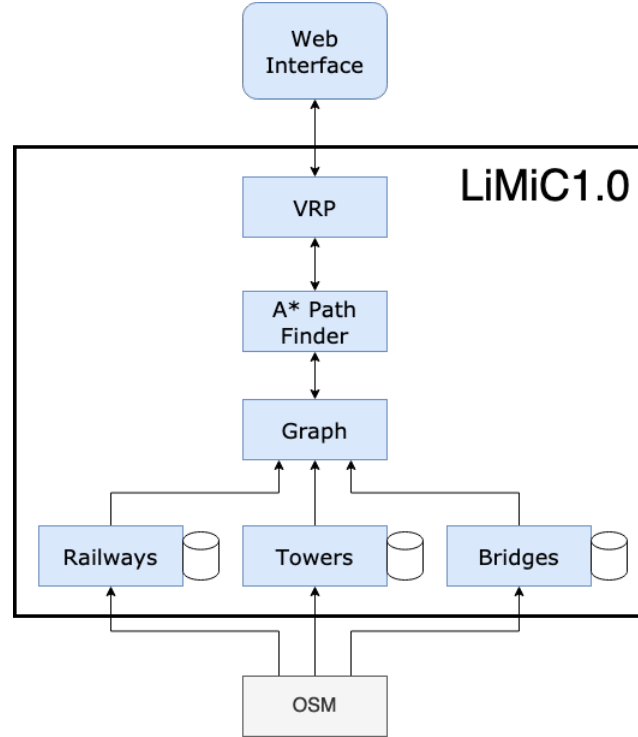


Figure 1: Application structure based on capabilities

much faster than Flask which was used for LiMiC monolith development [25]. It also provides asynchronous support crucial for taking advantage of microservice architecture. Asynchronous requests make the communication between routing solver and A* pathfinder much faster, especially when A* pathfinder is scaled. It enables concurrent computation of the shortest path.

Services are developed as follows:

- *Towers service* - the service uses Overpass API [26] to get the power towers and power lines data from OpenStreetMap. Service connects to the MongoDB Atlas database service and creates a database if it already does not exist. MongoDB Atlas provides a cloud database service for MongoDB databases. We chose MongoDB because it stores data in JSON-like documents and can be easily queried. Furthermore, it provides a geo-based search we use for finding indirect towers' neighbors needed for creating edges in the graph. Power towers are stored as a collection of objects with a unique identifier and location containing the tower's coordinates. Power lines are extracted using the same method and stored as a collection of objects with unique identifiers and an array of unique identifiers representing the nodes the line is passing through. Most of the nodes' unique identifiers are also towers', but they can also be a line intersections. For that reason, we created another collection of nodes contained in the power line arrays. Nodes' objects are defined the same as towers. Power line collection also contains tags with useful information on the number of cables, frequency, and voltage. The need for storing the power line information emerged because we want to find towers' direct neighbors. Tower's direct neighbor is the closest tower, or more of them, laying on the same power line. When creating the graph we want to give a higher cost to the paths between the indirect neighbors than direct since we want the UAVs to fly close to the infrastructure whenever that is

possible. Such a strategy is implemented for UAV flight regulations reasons. Tower service exposes two endpoints. One is providing the tower data and another one the data containing power lines. The data is used by the web interface to visualize the inspection targets. Graph service requests the data to create nodes and edges in the graph.

- *Railways service* - the service extracts railways location in the same way described in the towers service. It saves the data to the MongoDB database and sends it to the graph service on request.
- *Bridges service* - the service extracts polygons around the bridges and saves them to the database. Polygons are stored as the way types containing unique identifiers for nodes in the polygon. The nodes are stored in the same database containing the node's location. Bridges locations than can be combined with power line locations to create a graph and enable the UAVs to reach a bridge.
- *Graph service* - the service creates a graph based on the data received from either towers service or railways and bridges. The towers' locations represent the nodes in the graph while a pair of neighboring towers represent edges. The edge's cost is set up as the calculated distance between the neighboring towers. All the data for building the graph is requested from the towers service. The graph is created only once and it is used by A* pathfinder to determine the shortest path between the set of inspection targets and UAVs.
- *Routing solver service* - the service receives inspection targets and UAV locations and uses OR-Tools to determine which UAV should inspect which target. It creates asynchronous requests to the A* service to get the shortest path for each combination of UAV and target. Then, it builds the distance matrix and, based on path length, finds a near-optimal solution to the vehicle routing problem. It returns the path for each UAV as a set of waypoints containing locations in latitude and longitude.
- *A* pathfinder service* - the service requests a graph from the graph service and performs the A* algorithm on the targets and UAVs locations received from the routing solver service. A* algorithm is implemented from the NetworkX library. It returns the shortest path between a target and a UAV and returns the total path distance as well as the distance between each path segments with corresponding node locations.

4.3. Deployment Pipeline and DevOps tools

In order to deploy the application automatically every time the changes are made, we set up a pipeline using DevOps tools and technologies. Deployment pipeline consists of different stages for application building, deploying, and testing. We also enabled security scanning for code vulnerabilities and monitoring. To store the code, enable collaboration, and configure the pipeline, we used GitLab [27].

Microservices tend to be independent and isolated from each other. The most used and convenient way to isolate services is using container technology. Containers package up the application code and all its dependencies so the application can run quickly and reliably in any computing environment. Docker is the most popular and widely used container technology. Containers run from Docker images configured in Dockerfile. Docker image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries, and settings. Containers always run the same, regardless of the operating system and infrastructure. Compared with virtual machines, containers isolate application software from its environment but do not include a full copy of an operating system. Therefore, they take up less space and are much faster [28]. Each LiMiC1.0 service is a Python package containing multiple

modules and the Dockerfile. Docker images are built on the Python base image and run the application in a virtual environment after installing all the dependencies.

In LiMiC1.0, each service runs in a separated container and contains its Dockerfile. The whole process is configured using GitLab. Gitlab has built-in tools for software continuous integration and continuous deployment (CI/CD) which are used to run the pipeline. The codebase is hosted in the GitLab repository and contains a gitlab-ci.yml file describing the pipeline stages. Each service contains its yml file where the deployment and service port is configured. Every time the changes are pushed to the repository, GitLab runners execute scripts defined in the gitlab-ci.yml file and create a cluster deployments based on the service's yml file. We decided to automatize the LiMiC1.0 deployment to the server in the Kubernetes cluster as a staging environment. Figure 2 shows the application deployed to the Kubernetes cluster.

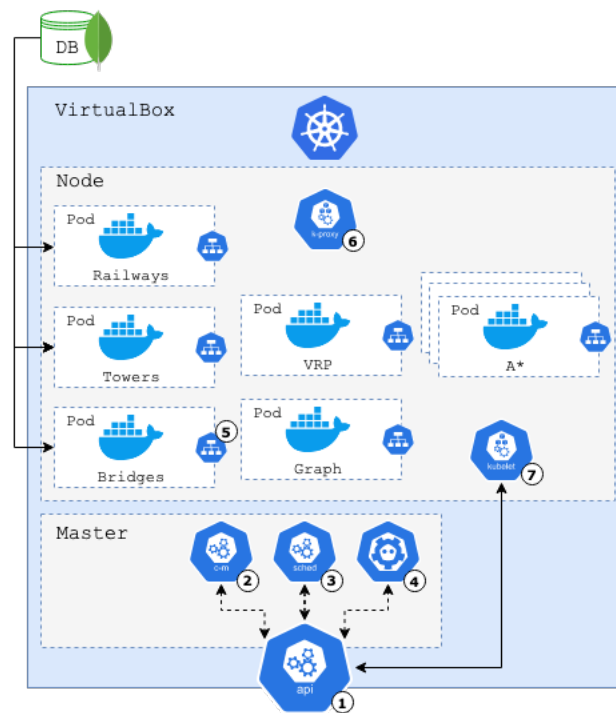


Figure 2: Application deployment in Kubernetes cluster

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. Kubernetes cluster consists of master and worker machines. Worker machines, also called nodes, run containerized applications in pods while the master manages workers and makes sure that the cluster is working in a configured way. The master node contains an API server which is the cluster entry point. API server ① runs a process that enables communication with the cluster. To visualize and manage all the cluster applications we can use the dashboard, as a web user interface, communicating with the API server. Otherwise, we can use API to communicate with the API server from the script or simply we can manage the cluster by writing commands in the terminal using a command-line tool. The master node keeps track of a cluster by running a controller manager. Controller manager ② is responsible for checking if all the nodes and pods are running and, if one goes down, the controller manager replaces it with a new one. Scheduler ③ decides on which node a newly created pod will run, based on the available and required resources. Etcd key-value storage ④ in the master node keeps status data about the

nodes. Master nodes and worker nodes communicate through a virtual network. The virtual network assigns an internal IP address to each pod. Pods communicate through the services ⑤ which contain permanent IP addresses. In case a pod restarts it will keep the service with the same address. Each service also provides a load balancing. Network rules for allowing the communication through the services are maintained by Kube proxy ⑥ running on each worker node. Kubelet ⑦ agent runs on each node assuring that containers run in pods as described in pods specifications. Container runtime is responsible for running the containers [29].

We decided to set up a Kubernetes cluster on a server running Linux Ubuntu 18.04.5 LTS using Minikube. Minikube allows you to set up a local cluster with the master process and worker process running on one node. We use it as a staging environment to deploy and test our application since the application development is still in an early stage and does not have any end users. Minikube creates a virtual machine VirtualBox on a local machine and the node runs in the VirtualBox. Kubectl is a command-line tool we use to interact with the cluster.

To automatize the deployment, we integrated the cluster with GitLab by providing a cluster API and token. To be able to run the CI/CD jobs, we installed the GitLab runner on the cluster. In the yml file, we configured three pipeline stages. In the first stage, the Gitlab runner runs a Docker container for each service based on the Docker image. Inside that container, we build a new image based on the corresponding service's Dockerfile and push it to the Gitlab container registry. The processes run in parallel for each service to speed up the creation of images. In the second stage, we create the deployments and services in the cluster based on the configuration file for each service. In the configuration file, we provide the path to the container registry where the images are stored in the previous stage and set the desired number of pod replicas for each deployment. We configure the ports where the services can be reached. We also enable monitoring through GitLab using Prometheus. The third stage is testing. We test if the services are responsive and if the returned result is correct. The test contains several inspection targets and UAV locations with calculated paths in between. The same targets and UAV locations are sent to the deployed system and the solution is compared with the correct one. The test is delayed for a minute after the deployment stage finishes successfully, to allow the Kubernetes to run all the pods following the yml configuration. The pipeline configuration file also includes a GitLab template for code vulnerability scanning i.e., Static Application Security Testing (SAST). The SAST checks for potentially dangerous attributes in a class, or unsafe code that can lead to unintended code execution and searches for vulnerabilities to cross-site scripting attacks that can be leveraged to unauthorized access to session data. The report can be downloaded after the pipeline executes correctly and the results are sorted by the priority of the vulnerability. In case we are merging changes from another branch to the master branch, GitLab will find the vulnerabilities in the new code that are different from the code in the master branch and report them. GitLab also offers visualization of the pipeline stages where we can see the progress and check the logs for the executed processes. The successful pipeline execution is shown in Figure 3. On every push to the GitLab repository, the code is scanned for the vulnerabilities, the services are containerized, automatically deployed to the Kubernetes cluster, and tested. The controller manager assures that desired number of pods is always running. Pod logs are available in GitLab for both GitLab-managed applications and services.

For monitoring the cluster and the application, we installed an open-source monitoring system Prometheus [30] through GitLab. For each service, we expose metrics endpoint using Prometheus middleware. In GitLab, we configured a custom dashboard to monitor



Figure 3: Application deployment pipeline

the number of requests coming to the services. The number of requests coming to the A* pathfinder is visualized in Figure 4. Requests visualized in red are coming from the routing solver while requests in blue are Prometheus requests for scraping metrics. We also monitor the percentage of failed requests. Since GitLab deprecated Prometheus and scheduled it for removal, it was not possible to add alerts. Otherwise, we would be able to add an alert to notify us when there is a certain percentage of failed requests. With Prometheus integration, we also monitor the cluster's CPU and memory usage as well as CPU, memory, and network metrics for each pod in the cluster. That dashboard is set up automatically with the installation of Prometheus.

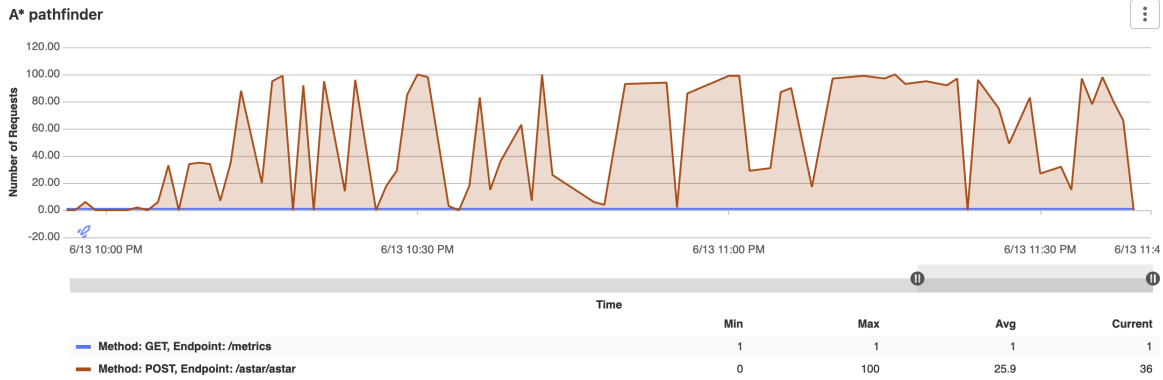


Figure 4: Requests to the A* path finder service

5. PERFORMANCE EVALUATION

To evaluate our application performance, we deployed it automatically through the GitLab CI/CD to the Minikube Kubernetes cluster. We compared the performance with the monolithic version of the application. To assure the same running environment, we created a Docker image for our monolith application and deployed it to the same cluster within a new namespace. We tested the performance by sending requests with different sets of sources and targets and measuring the time to process the requests. Sources represent UAV locations and targets are power tower locations. We chose the locations contained in the graph randomly. The data contains power tower locations in Denmark and we suppose that UAVs are located on the power tower locations on their start. We exponentially increment the number of sources from 1 to 16, and the number of targets from 1 to 64. One hundred requests were generated for each combination of sources and targets, thus 3500 requests in total. The same requests were made to both systems. Requests to LiMiC1.0 were made for a different number of pod deployments. First, we deployed only one pod per service and measured the processing time. Then we scaled the deployment with ten A* pathfinder

pod replicas to take the advantage of asynchronous communication between vehicle routing solver and A* pathfinder services. The processing time comparison between monolithic LiMiC and microservices LiMiC1.0 is shown in graphs in Figures 5 to 16. Processing time for microservices application without scaling is shown in green, with ten A* pathfinder pod replicas is shown in yellow, and the monolithic application processing time is shown in blue. Figures 5 to 9 show processing time for a different number of sources, i.e. UAVs, while altering the number of targets. Figures 10 to 16 show processing time for a different number of targets while altering the number of sources. Both options were visualized to provide a better analysis of the system's performance.

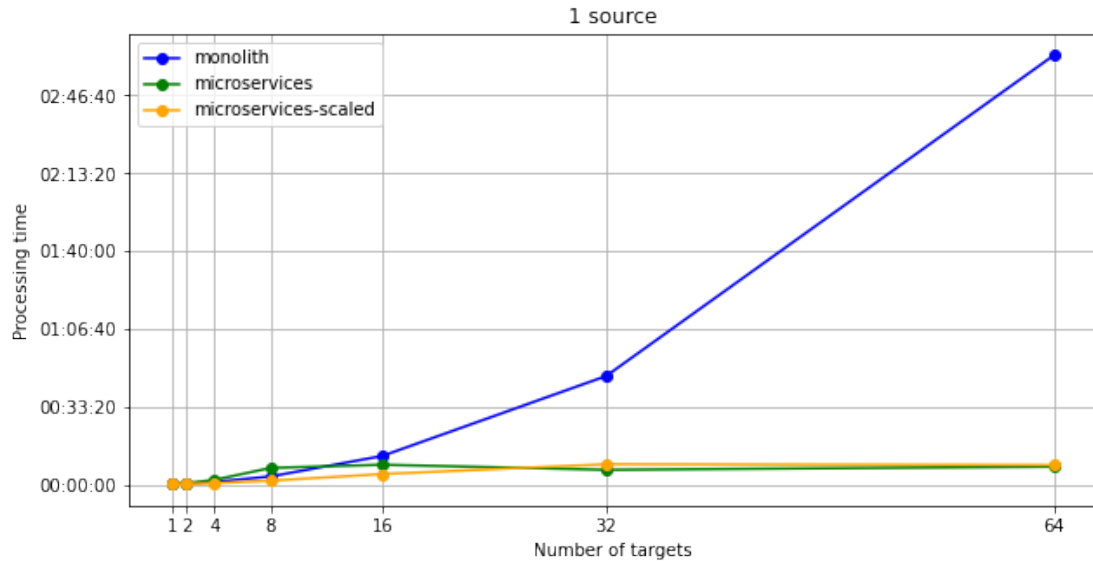


Figure 5: Processing time comparison between monolith and microservice application with 1 source and altering number of targets

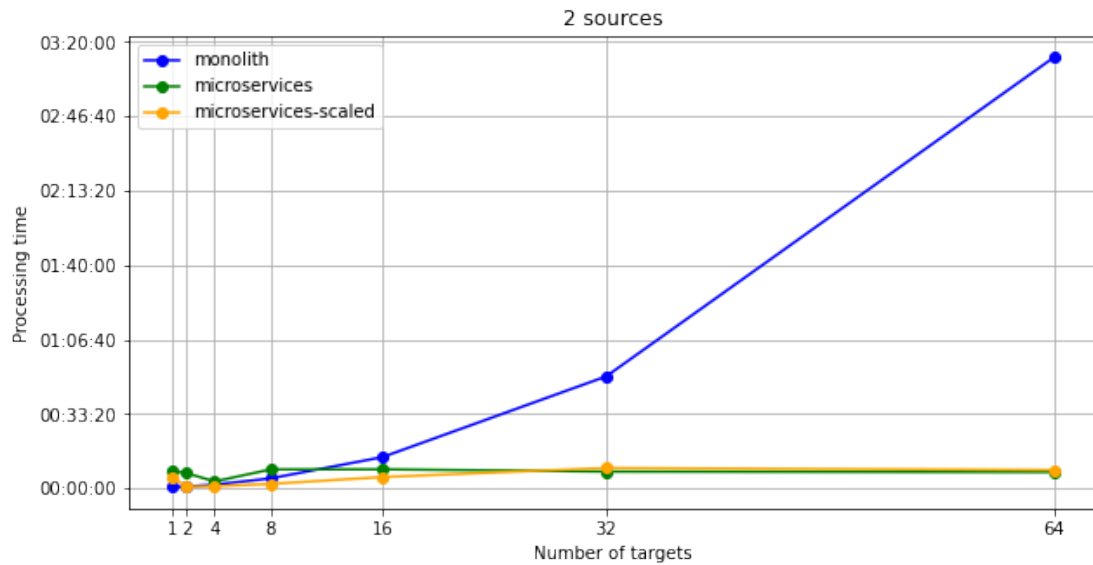


Figure 6: Processing time comparison between monolith and microservice application with 2 sources and altering number of targets

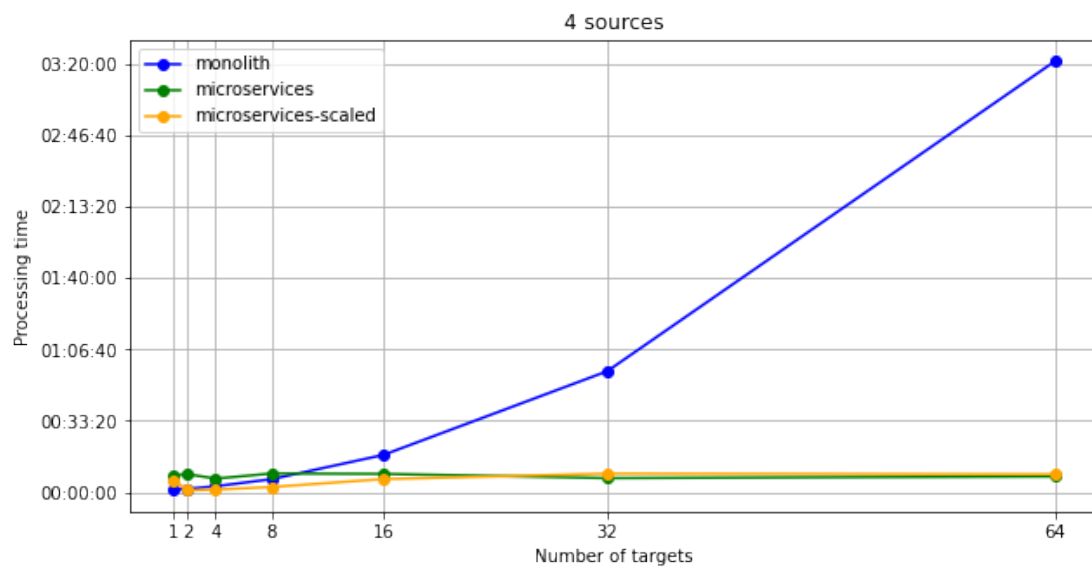


Figure 7: Processing time comparison between monolith and microservice application with 4 sources and altering number of targets

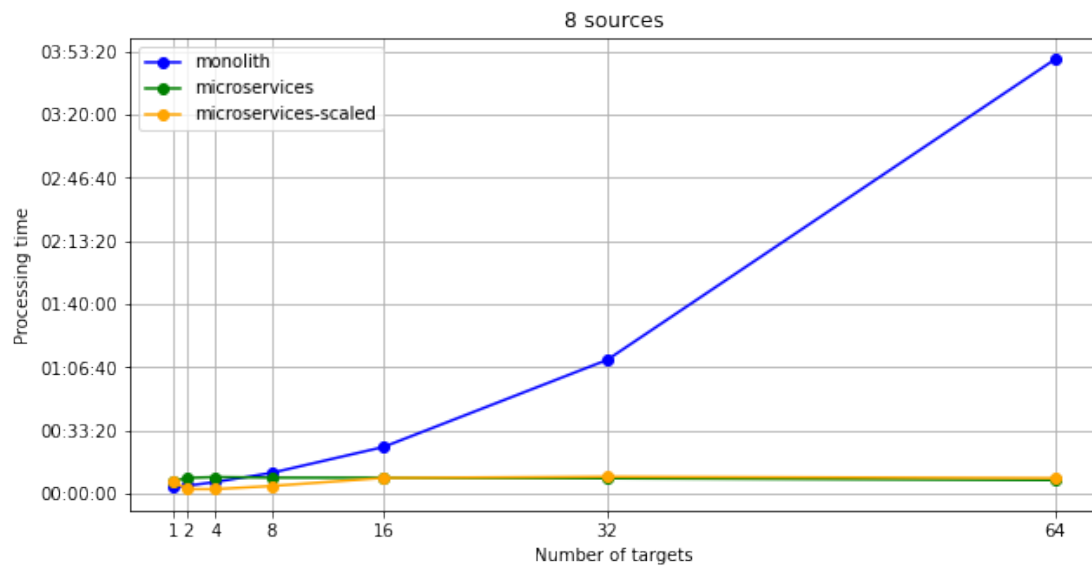


Figure 8: Processing time comparison between monolith and microservice application with 8 sources and altering number of targets

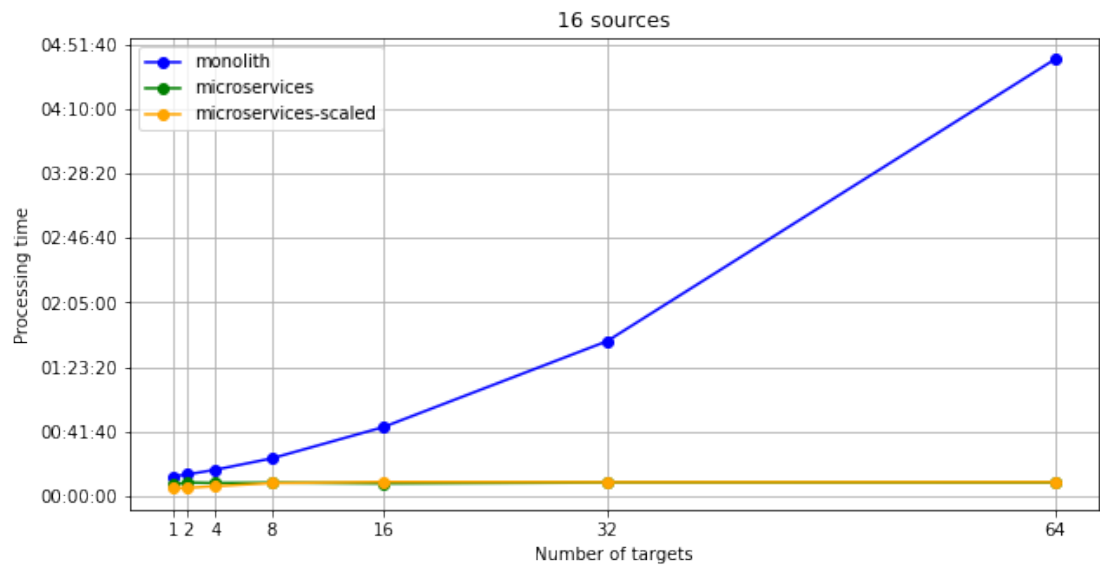


Figure 9: Processing time comparison between monolith and microservice application with 16 sources and altering number of targets

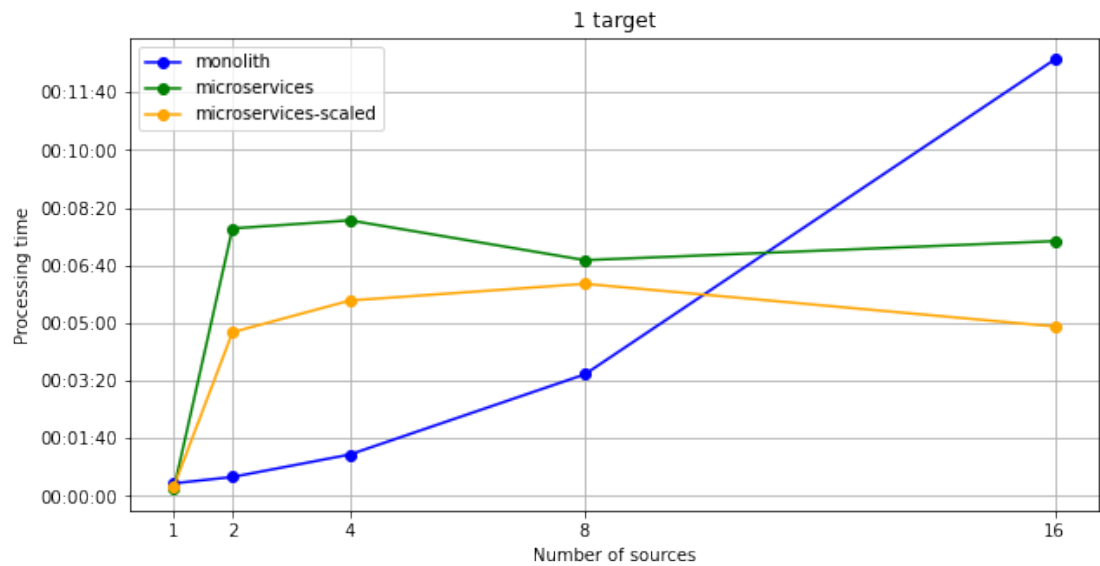


Figure 10: Processing time comparison between monolith and microservice application with 1 target and altering number of sources

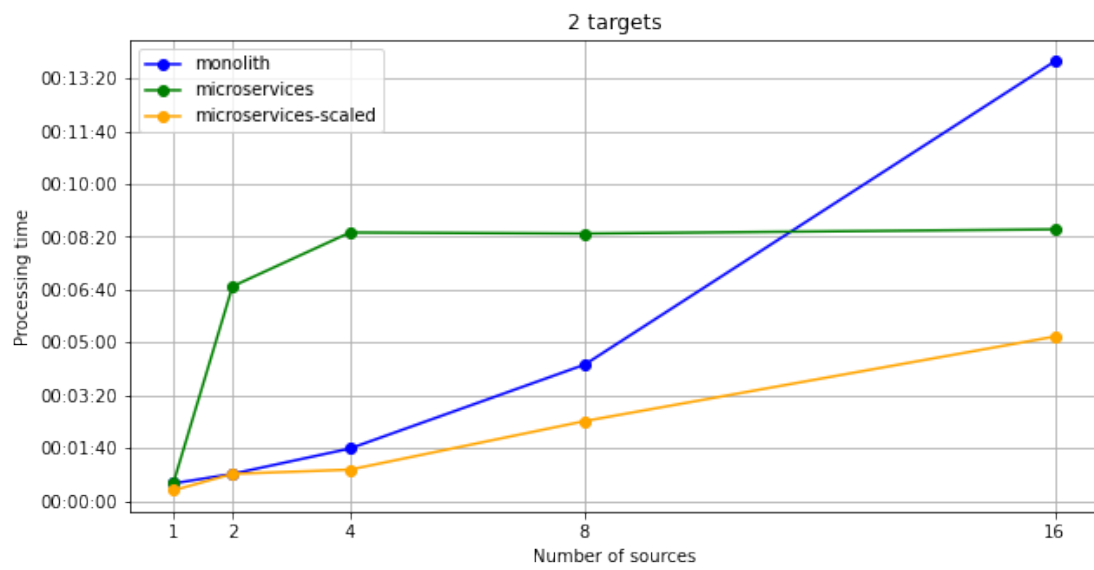


Figure 11: Processing time comparison between monolith and microservice application with 2 targets and altering number of sources

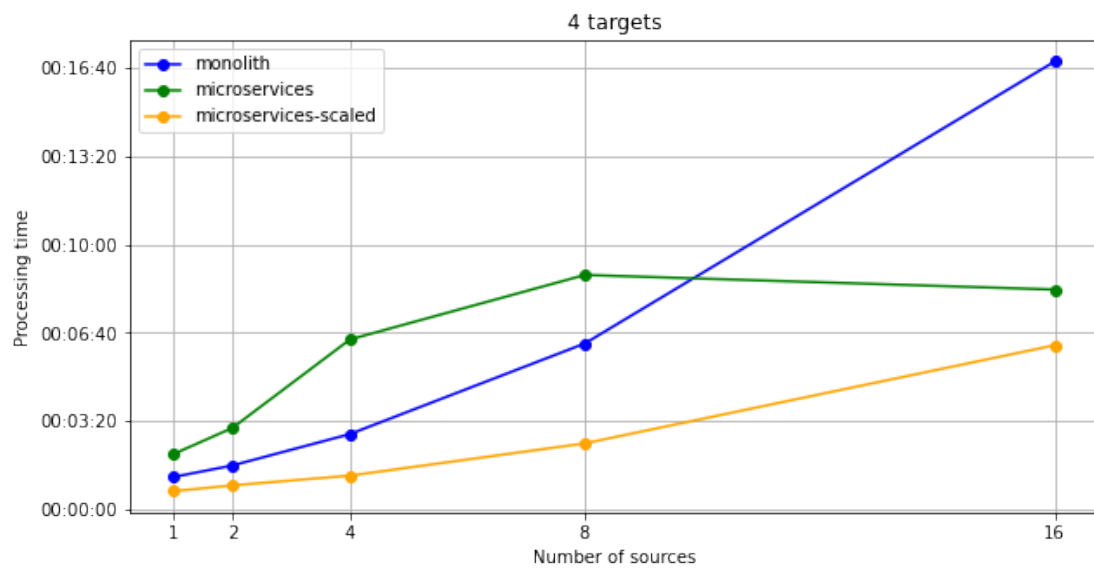


Figure 12: Processing time comparison between monolith and microservice application with 4 targets and altering number of sources

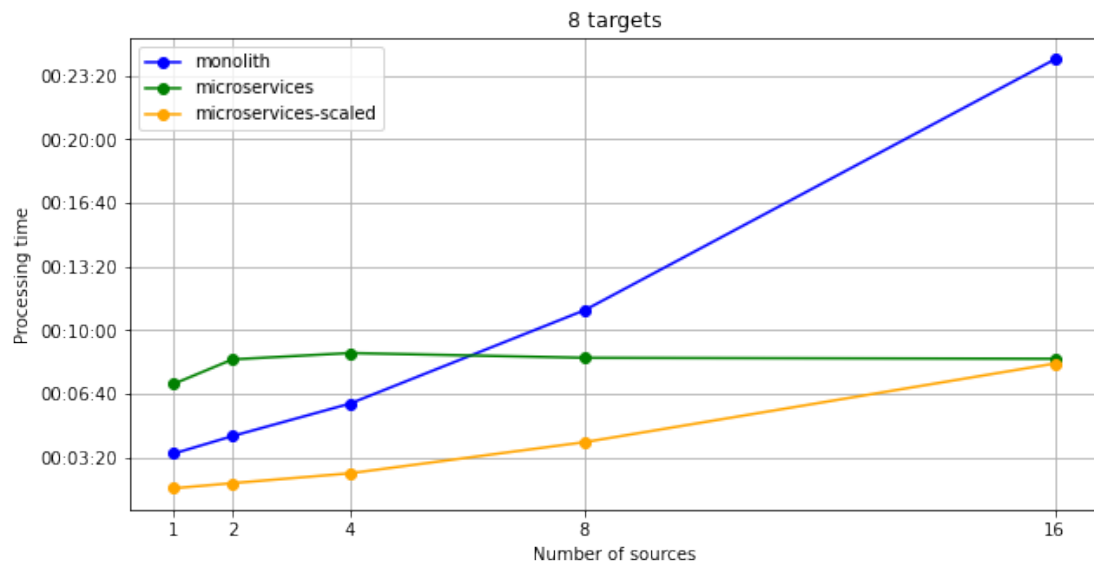


Figure 13: Processing time comparison between monolith and microservice application with 8 targets and altering number of sources

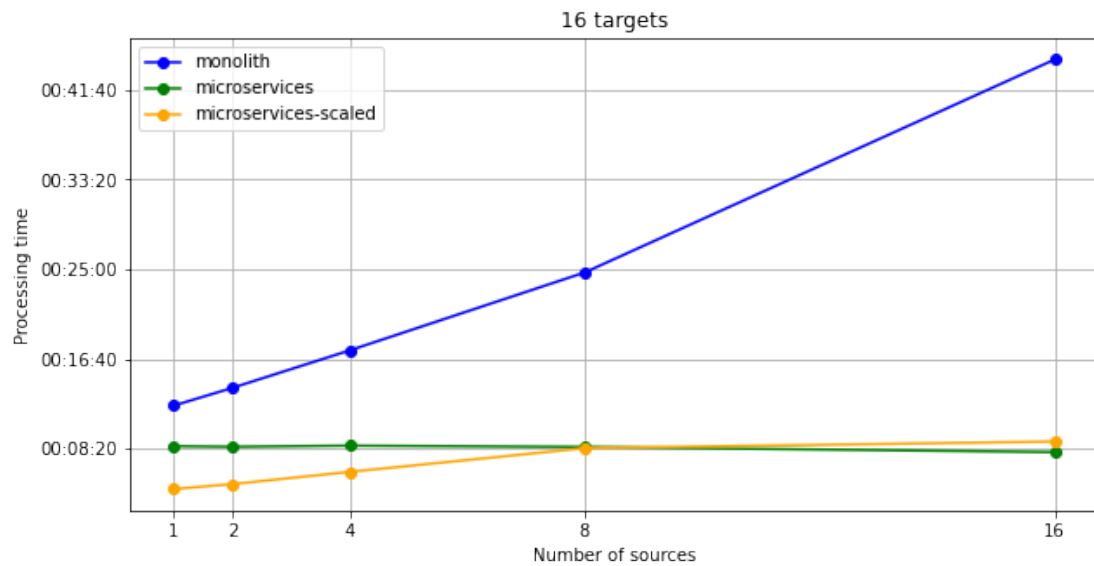


Figure 14: Processing time comparison between monolith and microservice application with 16 targets and altering number of sources

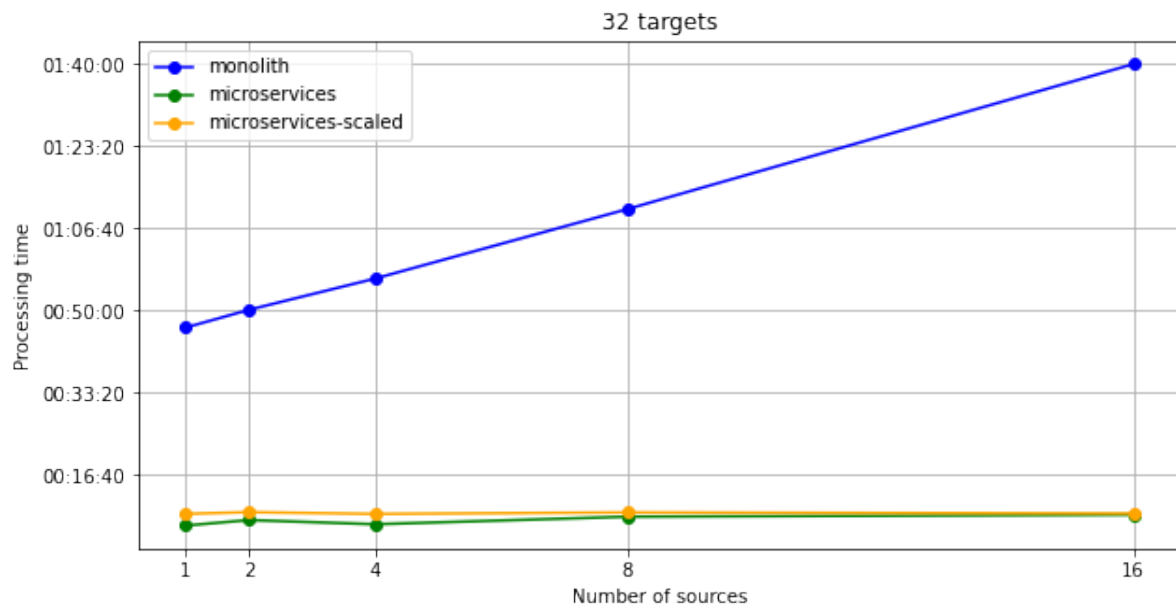


Figure 15: Processing time comparison between monolith and microservice application with 32 targets and altering number of sources

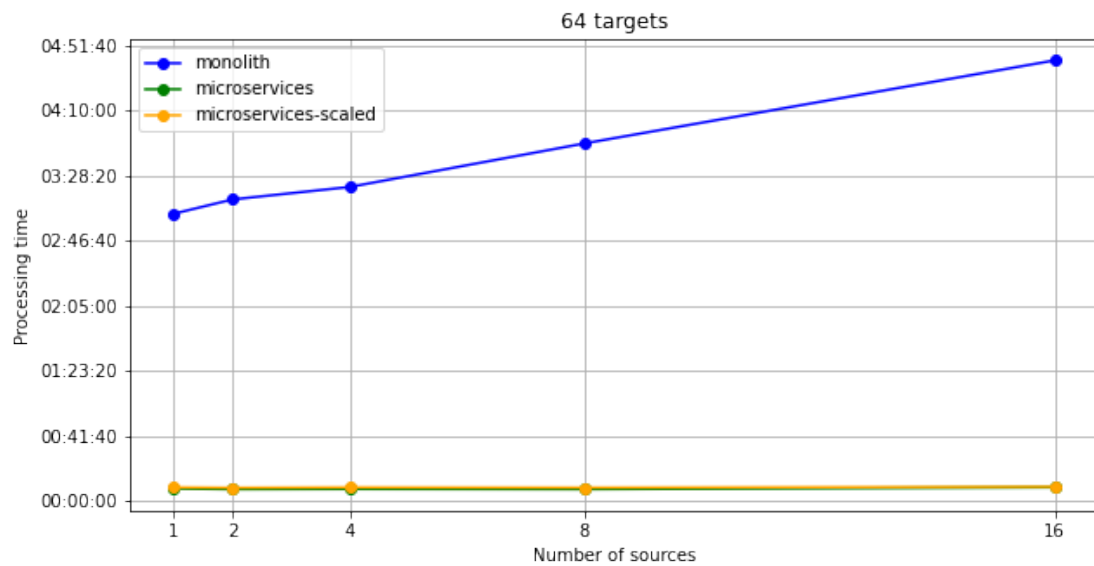


Figure 16: Processing time comparison between monolith and microservice application with 64 targets and altering number of sources

In all figures, we see that with the monolithic application, the processing time extends proportionally with the number of sources and targets while with microservices the processing time stabilizes and stops raising. For the microservices application, the overhead of internal communication between the services adds in processing time for two to eight sources visiting one target, which is best visible in Figure 10, where both scaled and not scaled microservice application performs slower than the monolith. The same is visible in Figures 6 to 8, where the path is calculated for sources to reach only one target. For a high number of sources, as in Figure 9, microservices perform faster for each number of targets. As the number of targets is raising, monolithic application processing time slows down, and microservices start outperforming the monolith, especially when scaled, as seen in Figures 11, 12, and 13. For 16 targets, LiMiC1.0 performs significantly better as shown in 14. By increasing the number of targets even more the processing time becomes near constant as seen in Figures 15 and 16. If we take the highest number of sources and targets and calculate the average time it takes to find near-optimal paths for 16 UAVs to visit 64 towers, we calculate that LiMiC1.0, when scaled, can process the request and serve the solution in about 5,32 seconds. Compared with LiMiC which takes 1 minute and 13 seconds on average. When scaled, LiMiC1.0 is slower than the monolithic LiMiC only when two, four, or eight UAVs are chosen for inspection of one target. However, it is unlikely for a user to pick either of the combinations mentioned, since most of the time it is intuitive when visualized, which UAV is the most suitable for the mission inspecting only one target. We also have to take into account that sources and targets are chosen randomly over the country of Denmark. Therefore they do not always represent a realistic set and sometimes it was not even possible for the solver to find a solution. We expect that the average processing time for a realistic set of sources and targets is slightly better. However, this experiment provided an insight into applications' performance and showed a positive impact of migrating the route calculation algorithm to microservices. The performance of the microservices application could be further improved by deploying to a more powerful cluster and taking the advantage of automatic scaling.

6. CONCLUSION & FUTURE WORK

To increase safety and reduce costs while inspecting the infrastructure, we developed an application for routing and scheduling UAVs to fly near the infrastructure. In this paper, we redesigned the application architecture to speed up future development, enable collaboration between developers, automate the deployment, and most important to speed up the route calculation. We implemented the application in microservice architecture and configured deployment pipeline stages using GitLab. We automatically trigger the Docker image creation, push the images to the container registry, scan the code for security vulnerabilities, deploy the application to the Minikube Kubernetes cluster and test the services' responsiveness. Also, we check if the results calculated are correct. We test the performance by comparing monolithic application and microservice application processing times. Microservices process requests faster, having the advantage of pod scaling and asynchronous communication between the services. Monolith application processing time exponentially rises when calculating paths for an increasing number of UAVs and power towers to inspect. By refactoring the calculation algorithm, the processing time is shorter and stabilizes after reaching a certain number of sources and targets.

To have a fully operational application for infrastructure inspection, we aim at implementing the additional services. In some areas, UAV flights are not permitted and we want to avoid these zones. While creating the graph, we will detect the infrastructure locations laying in the forbidden areas and reroute around them to assure flight safety. The application

has to be able to communicate with UAV and send the navigation waypoints calculated in the routing solver service. Therefore, we aim at implementing a message broker service to manage the communication between UAVs and the application. UAVs should be able to send data, including inspection images, through the same message broker. One of the additional services would estimate the UAVs' location to visualize their location on the web interface in every moment while the mission is in progress, even if the connectivity is not constant. We will implement storage for planned and executed missions to be able to retrieve the mission data after its execution. The application would also contain image analysis services to detect faulty infrastructure using machine learning models as well as image storage. Another important feature is a service for 3D infrastructure reconstruction which would be used for path planning around the power towers and bridges. An idea worth considering for future work would be to implement the weather prediction service and use it to determine if the weather in the area allows a safe flight. The microservice architecture we implemented and presented in this paper will allow us to add mentioned features easily as independent services. Concerning the deployment strategy, we aim at deploying the application to production to a multi-node Kubernetes cluster in the public cloud and test the application's load.

7. ACKNOWLEDGEMENTS

This project has received funding from European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 861111, Drones4Safety.

8. REFERENCES

- [1] Danielak M. Drones improve safety of infrastructure inspections. [Online] Available: <https://www.roadsbridges.com/drones-improve-safety-infrastructure-inspections>
- [2] Jordan S, Moore J, Hovet S, Box J, Perry J, Kirsche K, Lewis D, Tse ZT. State-of-the-art technologies for UAV inspections. IET Radar, Sonar & Navigation. 2018 Feb 8;12(2):151-64.
- [3] Mehrooz G, Ebeid E, Schneider-Kamp P. System design of an open-source cloud-based framework for internet of drones application. In 2019 22nd Euromicro Conference on Digital System Design (DSD) 2019 Aug 28 (pp. 572-579). IEEE.
- [4] Mehrooz G, Schneider-Kamp P. Optimal Path Planning for Drone Inspections of Linear Infrastructures. In GISTAM 2020 May 9 (pp. 326-336).
- [5] Selenium. [Online] Available: <https://www.selenium.dev/>
- [6] ul Haq S. Introduction to Monolithic Architecture and MicroServices Architecture. [Online] Available: <https://medium.com/koderlabs/introduction-to-monolithic-architecture-and-microservices-architecture-b211a5955c63>
- [7] Richardson C. What are microservices? [Online] Available: <https://microservices.io/>
- [8] Fowler M. Monolith first. [Online] Available: <https://martinfowler.com/bliki/MonolithFirst.html>
- [9] Tilkov S. Don't start with a monolith. [Online] Available: <https://martinfowler.com/articles/dont-start-monolith.html>
- [10] Richardson C. Pattern: Microservice Architecture. [Online] Available: <https://microservices.io/patterns/microservices.html>

ADAPTIVE FORGETTING, DRAFTING AND COMPREHENSIVE GUIDING: TEXT-TO-IMAGE SYNTHESIS WITH HIERARCHICAL GENERATIVE ADVERSARIAL NETWORKS

Yuting Xue¹, Heng Zhou^{1, 2}, Yuxuan Ding^{1*}, Xiao Shan¹

¹School of Electronic Engineering, Xidian University, Xi'an, China

²Institute of Systems Engineering, AMS, Beijing, China

ABSTRACT

The generation task from text to image generates cross modal data with consistent content by mining the semantic consistency contained in two different modal information of text and image. Due to the differences between the two modes, the task of text to image generation faces many difficulties and challenges. In this paper, we propose to boost the text-to-image synthesis through an adaptive learning and generating generative adversarial networks (ALG-GANs). First, we propose an adaptive forgetting mechanism in the generator to reduce the error accumulation and learn knowledge flexibly in the cascade structure. Besides, to evade the mode collapse caused by a strong biased surveillance, we propose a multi-task discriminator using weak-supervision information to guide the generator more comprehensively and maintain the semantic consistency in the cascade generation process. To avoid the refine difficulty aroused by the bad initialization, we judge the quality of initialization before further processing. The generator will re-sample the noise and re-initialize the bad initializations to obtain good ones. All the above contributions have been integrated in a unified framework, which is an adaptive forgetting, drafting and comprehensive guiding based text-to-image synthesis method with hierarchical generative adversarial networks. The model is evaluated on the Caltech-UCSD Birds 200 (CUB) dataset and the Oxford 102 Category Flowers (Oxford) dataset with standard metrics. The results on Inception Score (IS) and Fréchet Inception Distance (FID) show that our model outperforms the previous methods.

KEYWORDS

Text-to-Image Synthesis, Generative Adversarial Network, Forgetting Mechanism, Semantic Consistency.

1. INTRODUCTION

In the past few years, generative adversarial networks (GANs) [1] have boomed in the deep learning tasks. Various kinds of GANs [2], [3], [4] have brought amazing results on generating natural images through random noise. In order to generate images met the desire of users, conditional generative adversarial network (CGAN) [5] sets a condition as the target for the generator. deep convolutional generative adversarial network (DCGAN) [6] combines GANs with convolutional neural network (CNN) to improve the quality of generated images. To incorporate more accurate surveillance, auxiliary classifier generative adversarial network (ACGAN) [7] requires the discriminator to output both probability and fine-grained category.

Although significant progress has been made in generating visually realistic images, generating images that match the given text descriptions is still challenging. The conditional align Deep Recurrent Attention Writer (alignDRAW) [8], which is the first text-to-image generation model extending DRAW [9] to generate images from texts. However, the synthesized images are blurred with a low-resolution of 36x36. Then, Reed et al. [10] introduce GAN to text-to-image task, which follows DCGAN and CGAN to generate images from texts. As the training process is not stable, GAN-INT-CLS only generates plausible images for birds and flowers. To reduce the unstable of the training process, the popular text-to-image generation methods [11], [12], [13] mainly apply a multi-stage generator to supplement more restrictions. However, there are still three issues in the multi-stage structure. First, the cascade structure accumulates the incorrect and redundant information during the generation process. Second, the output of discriminators is the probability of reality which cannot guide the generator comprehensively. Finally, bad initialization of images has unclear parts which make the refine difficulty.

To address these issues, we propose a new text-to-image synthesis model called ALG-GAN. In the process of cognition, we ignore the redundant and incorrect information to learn and summarize knowledge more efficiently. Inspired by this, we incorporate a pair of down-sampling and up-sampling convolutional layers to the up-block to bring in the forgetting process. Thus, the model has opportunity to throw the useless and improper information away, which is called forgetting mechanism (FM). For the second problem, we design a multi-task discriminator (MTD) to fully utilize the additional weak-supervision information in the training process, which can help the discriminator to guide the generator in detail. For the last problem, multi-stage generators start from the initial images to synthesize larger images, regardless the quality of initial images. However, people do not perform in this way. Good painting usually starts from a satisfied draft. Thus, we propose to supervise the generation process of small image to guarantee its quality. It is called drafting mechanism (DM).

The main contributions of this paper are summarized as follows:

- Forgetting mechanism: we propose a down-up sampling dual structure, which allows the network to forget information during the generation process.
- Comprehensive guiding discriminator: it guarantees the comprehensive guidance by additional weak-supervision.
- Drafting mechanism: we supervise the generation process through discriminator to guarantee the quality of initialization.

We conduct experiments to evaluate the proposed ALG-GAN model on the Caltech-UCSD Birds 200 (CUB) dataset [14] and the Oxford 102 Category Flowers (Oxford) dataset [15]. The quality of generated images is measured using the inception score (IS) [16] and the Fréchet inception distance (FID) [17]. The experimental results indicate that our ALG-GAN model performs better than the state-of-the-art text-to-image synthesis methods. On CUB, we improve IS from 4.36 to 4.62. FID decreases from 16.899 to 16.500. On Oxford, the result of IS is 4.10. FID achieves 44.307. It proves that our model generates more realistic images.

The remainder of this article is organized as follows. In Section II, the related works of image generation are introduced from two aspects: multi-stage generator and multi-task discriminator. In Section III, we introduce AttnGAN as the baseline of our model. Section IV, the text-to image generation model ALG-GAN we proposed is introduced in detail. ALG-GAN mainly includes three parts: an adaptive forgetting and drafting generator and a comprehensive guiding discriminator. In Section V, compared with the state-of-the-art methods on the public dataset, it shows that this method has superior performance. The effectiveness of ALG-GAN is proved by a large number of ablation experiments. The conclusions and future work are shown in Section VI.

2. RELATED WORKS

Our work mainly refers to the hierarchical structure of generator and the auxiliary task of discriminator. I will introduce the related work from the following two aspects.

2.1. Multi-Stage Generator

In the text-to-image task, Reed et al. [10] propose a structure called GAN-INT-CLS based on CGAN and DCGAN, which uses sentence embedding as the condition to generate images. Since it is difficult to control the generation process, generated images are small with 64×64 resolution which lack details and easily suffer from mode collapse. To address the size limitation, Zhang et al. [11], [12] extend single-stage to multi-stage in their StackGAN and StackGAN++. The stacking structure makes the generation process more controllable to synthesize large images. Considering that the word level information can guide the local information generation, Xu et al. [13] introduce AttnGAN with deep attentional multimodal similarity model (DAMSM) to refine images which takes into account of both local and global information. However, the multi-stage structure has pros and cons. On the one hand, stacking structure addresses the size limitation by means of bit by bit surveillance. We also incorporate drafting mechanism in the stack structure to guarantee a better initialization. On the other hand, multi-stage incorporates restrictions in each stage. However, the biased surveillance from the discriminator at each stage will be accumulated. We propose a forgetting mechanism to promote the model learn adaptively. Meanwhile, the model is prone to collapse when it is supervised by a biased strong surveillance. Thus, building a comprehensive surveillance is also crucial for the model to be success.

2.2. Multi-Task Discriminator

The original discriminator only judges the reality of input images. To guide the generator better, Augustus et al. [7] propose ACGAN by adding an auxiliary classifier to the discriminator of GAN, which achieves the start-of-the-art results. It confirms that additional classification task in the discriminators can guide a better generator. Following ACGAN, Ayushman et al. [18] propose TAC-GAN where the auxiliary classifier classifies the category of birds. It obtains good results. Following TAC-GAN, Cha et al. [19] propose Text-SeGAN adding a semantic classifier to guarantee the semantic consistency. Following TAC-GAN, we assume that not only category label, but also some other fine-grained weak-supervision information like attributes also meet this end.

3. THE PRELIMINARY: ATTENTION BASED HIERARCHICAL GENERATIVE ADVERSARIAL NETWORK

The text-to-image model AttnGAN [13] consists of an attention based hierarchical generator G and a discriminator D . G has two main components: The initialization and DAMSM based up-block. In the initialization, firstly, the input text description is transformed by a text encoder into the word-level representations and a global feature, which is used as the sentence condition. Then, G predicts the rough sketch of image I_0 according to a random noise vector with the sentence condition after conditioning augmentation (CA). The noise vector is normally distributed. After initialization, more fine-grained visual contents are supplemented to the initial image by up-block, which makes it more photo-realistic. D distinguishes not only the real data from synthesized images, but also the matched sentence conditions from mismatched conditions. During training, G and D are following the two-player min-max game with value function : $V(G, D)$:

$$\begin{aligned}
\min_G \max_D V(D, G) = & E_{I \sim p_{\text{data}}} [\log D(I)] + E_{I \sim p_{\text{data}}} [\log D(I, \bar{e})] \\
& + E_{\hat{I} \sim G} [\log(1 - D(\hat{I}))] + E_{\hat{I} \sim G} [\log(1 - D(\hat{I}, \bar{e}))] \\
& + L_{CA} + L_{DAMSM}
\end{aligned}$$

where E means the expectation. $D(I)$ and $D(\hat{I})$ compute the probability of reality, while $D(\hat{I}, \bar{e})$ and $D(I, \bar{e})$ compute the probability of matching between text and image. L_{CA} represents the K-L divergence between the standard Gaussian distribution. L_{DAMSM} measures the correlation between images and corresponding text descriptions. Both L_{CA} and L_{DAMSM} are only related to G .

4. THE PROPOSED ALG-GAN

Our text-to image generation model ALG-GAN is demonstrated in Figure 1, which consists of an adaptive forgetting and drafting generator and a comprehensive guiding discriminator.

4.1. Adaptive Forgetting and Drafting Generator

To address the cons of hierarchical structure, we incorporate two new mechanism in our generator.

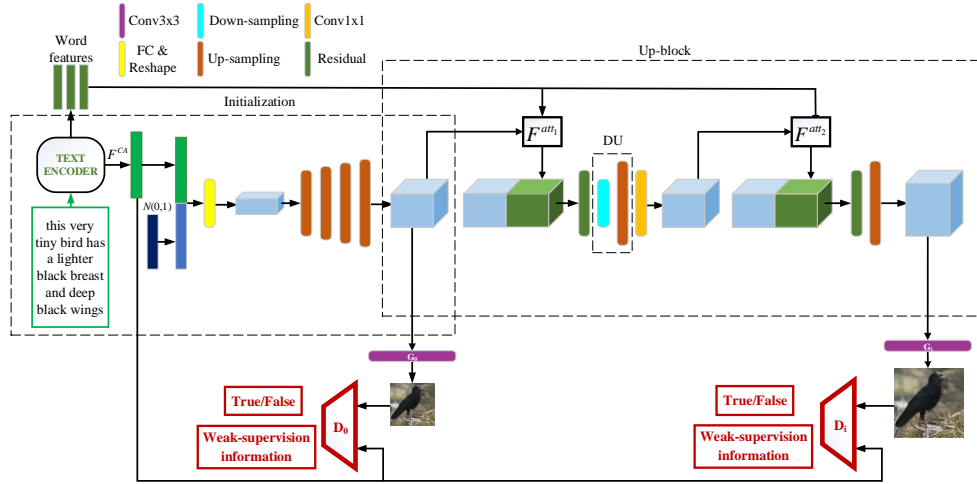


Figure 1: The architecture of the proposed ALG-GAN

Forgetting Mechanism To handle the redundant and incorrect information during the generation process, we propose a novel up-block module, which utilizes a down-up sampling dual structure (DU) to learn information adaptively.

Redundant information arises in the feature map after we first refining images with DAMSM $F_i^{att_1}$ in the up-block. So, we utilize the down-sampling operation to forget some information. To keep the size, we combine the down-sampling operation with an up-sampling operation. Then, we utilize the second DAMSM $F_i^{att_2}$ to enhance the learnt knowledge. The new image features are obtained by:

$$\begin{aligned}
h_i^{temp} &= F_{DU} \left(h_i, F_i^{att_1} (e, h_i) \right) \\
h_i^{new} &= \phi \left(h_i^{temp}, F_i^{att_2} (h_i^{temp}, e) \right)
\end{aligned} \tag{2}$$

where h_i is image feature, $F_{DU}(\cdot)$ represents down-up sampling operation, $\phi(\cdot)$ is implemented as a residual module with 1×1 convolution to adjust the number of channels.

Drafting Mechanism In view of the painting process of human being, generator just likes a painter, guided by the discriminator. Actually, even professional painters still make drafts. They will judge their draft before they draw further. Such a judgment is also necessary in our generation process. However, it is difficult for the generator to judge the quality of the initialized image. We use the discriminator to conduct this task. In our DM, when the initialization of the image does not reach the judgment, the model will re-sample the noise and re-initialize the image until the requirements are met. Compared with current works, we are the first one who supervises the generation process.

See Figure 2, our generation process is defined as follows:

$$\begin{aligned}
h_0 &= re \left(F_0 \left(z, F^{CA}(\bar{e}) \right) \right) \text{ if } CE \left(D_0^{fe} \left(\hat{I}, \bar{e} \right), 1 \right) > \alpha \\
h_i^{temp} &= F_{DU} \left(h_i, F_i^{att_1} (e, h_i) \right) \\
h_i^{new} &= \phi \left(h_i^{temp}, F_i^{att_2} (h_i^{temp}, e) \right) \\
h_{i+1} &= F_{i+1} \left(h_i^{new} \right) \\
I_i &= G_i (h_i), \text{ for } i = 0, 1, 2, \dots, m-1
\end{aligned} \tag{3}$$

where F^{CA} means CA operation. $re(\cdot)$ donates re-sampling the noise and re-initializing the image. When the cross entropy $CE(\cdot)$ between the output of final epoch discriminator $D_0^{fe}(\hat{I}_0, \bar{e})$ and real label is larger than α , the model will re-initialize the image.

4.2. Comprehensive Guiding Discriminator

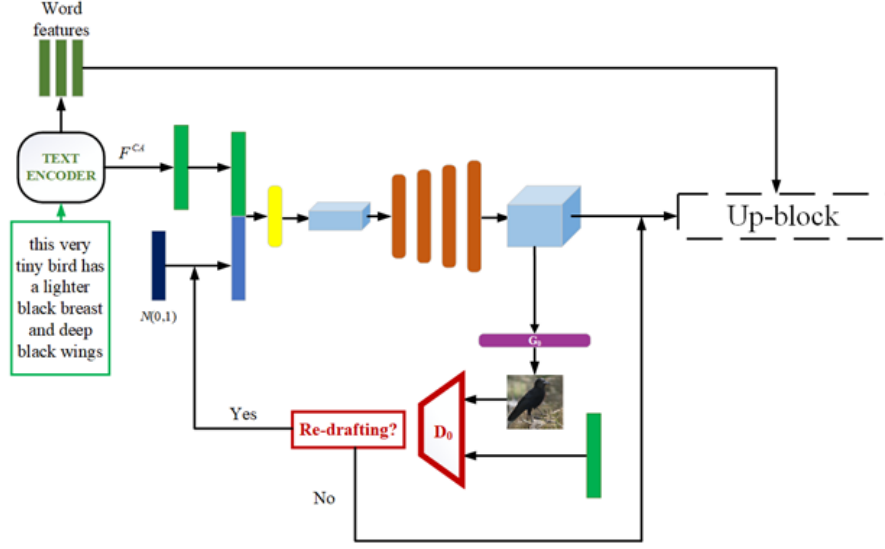


Figure 2: The generation process of ALG-GAN. We apply the final epoch discriminator to conduct the re-drafting judgment.

As mentioned in Preliminaries, the traditional text-to-image discriminators only compute the probability of reality and matching, which cannot guide the generator comprehensively. To address this, we propose a comprehensive guiding discriminator to guarantee the semantic consistency on the generated image during the generation process through the additional weak-supervisions such as category label, the fine-grained attribute label etc. The comprehensive guiding discriminator is shown in Figure 1. It has an auxiliary task of predicting the weak-supervision information on the synthesized and real data. Thus, discriminators give comprehensive surveillance rather than a biased one on the generator. The outputs of discriminators are defined as

$$\begin{cases} p_u = D_i^{GAN}(I) \\ p_c = D_i^{GAN}(I, \bar{e}) \\ p_{ws} = D_i^{ws}(I, \bar{e}) \end{cases} \quad (4)$$

where D is implemented as a discriminator. p_u means predicting the image whether it is real or synthesized. p_c measures the matching of image and sentence. p_{ws} discriminates the additional weak-supervision information on images.

4.3. Objective Function

We define our objective function as following.

$$L = \sum L_i^{GAN} + \lambda_1 L_{CA} + \lambda_2 L_{DAMSM} + \lambda_3 \sum L_i^{ws} \quad (5)$$

In Eq.(5), L_i^{GAN} is an adversarial loss. It is related to GAN [1]. λ are the corresponding weights of CA loss L_{CA} , DAMSM loss LDAMSM and weak-supervision loss L_{ws} . The definition of L_{CA} follows StackGAN++ [12]. The definition of L_{DAMSM} follows AttnGAN [13]. L^{ws} is defined as Eq.(6), which guarantees the comprehensive surveillance of discriminator.

$$L_i^{ws} = CE(p_{ws}, l_{re}) \quad (6)$$

In Eq.(6), l_{re} is real label of weak-supervision information. $CE()$ is implemented as cross entropy. The discriminators minimize the training objective function L_D , which is defined as follows. The subscript denotes the index of the discriminator. Table 1 shows the inputs to the discriminators.

$$L_{D_i} = L_{D_i}^{GAN} + \lambda_3 L_i^{ws} \quad (7)$$

$$L_{D_i}^{GAN} = -\frac{1}{2} E_{I_i \sim p_{data}} [\log D_i(I_i)] - \frac{1}{2} E_{\hat{I}_i \sim p_{G_i}} [\log(1 - D_i(\hat{I}_i))] \\ - \frac{1}{3} E_{I_i \sim p_{data}} [\log D_i(I_i, \bar{e})] - \frac{1}{3} E_{\hat{I}_i \sim p_{G_i}} [\log(1 - D_i(\hat{I}_i, \bar{e}))] \\ - \frac{1}{3} E_{I_i \sim p_{data}} [\log(1 - D(I_i, \hat{e}))] \quad (8)$$

$$L_i^{ws} = CE(D_i^{ws}(I, \bar{e}), l_{re}) + \frac{1}{2} CE(D_i^{ws}(\hat{I}, \bar{e}), l_{re}) + \frac{1}{2} CE(D_i^{ws}(I, \hat{e}), l_{re}) \quad (9)$$

The first row of $L_{D_i}^{GAN}$ means the unconditional loss, which distinguishes the real images from synthetic images. Another row is the conditional loss, which measures whether the text and image are matching. Notice that discriminators should minimize the matching probability of real image with mismatching text pair.

Table 1. The meaning of inputs for the discriminators.

Input	Meaning
I	Real image
\hat{I}	Generated image
(I, \bar{e})	Real image with matching text
(\hat{I}, \bar{e})	Generated image with corresponding text
(I, \hat{e})	Real image with mismatching text

When we optimize the generators, there are no real or mismatched images for discriminators to be fed as input. The generators should minimize the unconditional loss and conditional loss $L_{G_i}^{GAN}$

to synthesize the real and matching images. The subscript denotes the index of the generator. The training objective L_G is defined as follows.

$$\begin{aligned}
 L_{G_i} &= L_{G_i}^{GAN} + \lambda_1 L_{CA} + \lambda_2 L_{DAMSM} + \lambda_3 L_i^{ws} \\
 L_{G_i}^{GAN} &= -\frac{1}{2} \mathbb{E}_{\hat{I}_i \sim p_{G_i}} \left[\log \left(D_i(\hat{I}_i) \right) \right] - \frac{1}{2} \mathbb{E}_{\hat{I}_i \sim p_{G_i}} \left[\log \left(D_i(\hat{I}_i, \bar{e}) \right) \right] \\
 L_i^{ws} &= CE \left(D_i^{ws} \left(\hat{I}_i, \bar{e} \right), l_{re} \right)
 \end{aligned} \tag{10}$$

5. EXPERIMENTS

We conduct extensive experiments to validate ALG-GAN. First, we compare our ALG-GAN with the state-of-the-art GAN models [10], [11], [12], [13], [18], [20], [21]. Then, we validate the effectiveness of each new module proposed by our method, including FM, DM and MTD.

5.1. Experimental Setup

Dataset We use CUB [14] and Oxford [15] datasets to verify the text description based image generation. We preprocess and split the images into two disjoint sets following the same pipeline as GAN-INT-CLS [10]. CUB contains 11,788 bird images belonging to 200 categories, where 150 categories with 8,855 images are employed for training while the remaining 50 categories with 2,933 images are used for testing. Besides, CUB contains category and fine-grained attribute annotations. We choose them as the weak-supervision information and figure out the effectiveness for each of them. Oxford contains 8,189 images of flowers from 102 different categories, where 82 categories with 7,034 images are employed for training while the remaining 20 categories with 1,155 images for testing. Oxford only has category annotation, which is employed as the weak-supervision information. Each image in both CUB and Oxford has 10 text descriptions.

Parameter and model setting In our experiments, same as the setting [13], we define λ_1 as 1.0 and λ_2 as 5.0 following. We define λ_3 as 1.0 empirically. For text embedding, we employ a pretrained text encoder on CUB. For Oxford, we train the text encoder. During training, we fix the parameters of encoder to get the word features and sentence features. Then, we train AttnGAN [13] for Oxford as the baseline.

Evaluation metric We use IS [16] and FID [17] as the quantitative evaluation measures. IS measures both quality and diversity of generated images. It computes KL-divergence between the generated class distribution and the real class distribution, which uses the pre-trained Inception v3 network. A higher score means a better performance. FID computes the Fréchet distance between generated images and real images using the extracted features from a pre-trained network. A lower FID means a closer distribution between generated images and real ones.

5.2. Comparative Results

We compare our results with the state-of-the-art text-to-image methods on CUB [14] and Oxford [15] datasets. We report the results of IS in Table 2. ALG-GAN outperforms other methods with a higher IS. It indicates that ALG-GAN generates images with better quality and diversity.

Table 2. The comparison of IS by our ALG-GAN and the state-of-the-art GAN models on CUB and Oxford datasets

Methods	CUB	Oxford
GAN-INT-CLS	2.88±0.04	2.66±0.03
TAC-GAN	-	3.45±0.05
StackGAN	3.70±0.04	3.20±0.01
StackGAN++	4.04±0.06	-
AttnGAN	4.36±0.03	3.74±0.09
HDGAN	4.15±0.05	3.45±0.07
MirrorGAN	4.56±0.05	-
Our (ALG-GAN)	4.62±0.07	4.10±0.08

Table 3. FID between AttnGAN and ours, lower is better

Methods	CUB	Oxford
Baseline (AttnGAN)	16.898	46.459
Our (ALG-GAN)	16.500	44.307

Table 3 compares the performance between AttnGAN and ALG-GAN with respect to FID on CUB and Oxford. We measure FID by the officially pre-trained model. After resizing the real test images and the synthesized images in the same size, we compute FID between them. Compared with AttnGAN, our ALG-GAN decreases FID from 16.898 to 16.500 on CUB and from 46.459 to 44.307 on Oxford, which demonstrate that ALG-GAN can learn a better data distribution on objects. Representative examples generated from text descriptions by different methods are shown in Figure 3.



Figure 3. Qualitative examples of the proposed ALG-GAN comparing with HDGAN [20] and AttnGAN [13] on CUB and Oxford dataset.

5.3. Ablation study and discussion

To further demonstrate the effectiveness of each component, we perform some ablation experiments. Table 4 shows the results. These results demonstrate that each component in ALG-GAN is indispensable.

Table 4. IS produced by combining different components of ALG-GAN.

Methods	CUB	Oxford-102
Baseline	4.36±0.03	3.74±0.09
Baseline + FM	4.46±0.04	3.86±0.08
Baseline + FM + MTD(CCT)	4.56±0.04	4.08±0.08
Our (Baseline + FM + MTD(CCT)+DM)	4.58±0.04	4.10±0.08
Baseline + FM + MTD(ACT)	4.59±0.05	-
Our (Baseline + FM + MTD(ACT)+DM)	4.62±0.04	-

FM Baseline + FM improves IS of 2.3% over the baseline on CUB. Meanwhile, It results in 3.2% improvement on Oxford. The results confirm that forgetting the redundancy and incorrect information benefits the generation. In addition, in order to prove that the improvement of model performance is not caused by the increase in computing power after the introduction of the new structure, we set up a comparative experiment to replace the down-up sampling dual structure in ALG-GAN with naive Conv3×3. IS is shown in Table 5. It can be seen from the results that the introduction of additional convolutional layers will cause the model learn knowledge more redundant, resulting in a decrease performance.

CGD Based on FM, we further evaluate the effectiveness of CGD by validate Baseline + FM + CGD. When the discriminators conduct the category classification task (CCT), it improves IS from 4.46 to 4.56 on CUB and 3.86 to 4.08 on Oxford. When the discriminators conduct the attribute classification task (ACT), it improves IS from 4.46 to 4.59 on CUB. Those improvements prove that the weak-supervision of discriminator guides generators more comprehensively. Moreover, more fine-grained guidance results in better quality of the generated images. Because of the lack of attribute annotation in Oxford, we do not verify the validity of attribute classification on the flower dataset.

Table 5. The comparison of IS by FM and naive Conv3×3 on CUB and Oxford datasets

Method	CUB(ACT)	Oxford(CCT)
Baseline	4.36 ± 0.03	3.74 ± 0.09
Baseline + FM + CGD	4.59 ± 0.05	4.08 ± 0.03
Baseline + Conv3×3 + CGD	4.33 ± 0.05	3.64 ± 0.07

DM We discuss the effect of hyper parameter α in DM to IS through Baseline + FM + CGD. When α is set as 5.1, the model has best performance on ACT. When the weak-supervision information is category label, $\alpha = 5.6$ achieves best results. On Oxford, IS is stable when α are set from 5.5 to 5.7. The results of IS are shown in Table 6 and 7, which show that guaranteeing the initialization quality through supervision on the generation process benefits the subsequent image refinement. However, although DM works well, it is difficult to find one α to apply to all models. The reason is that α is influenced by the model initialization and other hyper parameters such as learning rate and batch size etc.

Table 6. Results on CUB from different α in DM

Method	CUB
Baseline + FM	4.46±0.04
Baseline + FM +MTD(ACT)	4.59±0.05
Baseline + FM +MTD(ACT) +DM ($\alpha = 4.5$)	4.57±0.05
Baseline + FM +MTD(ACT) +DM ($\alpha = 5.0$)	4.60±0.05
Baseline + FM +MTD(ACT) +DM ($\alpha = 5.1$)	4.62±0.04
Baseline + FM +MTD(ACT) +DM ($\alpha = 5.2$)	4.60±0.06
Baseline + FM +MTD(ACT) +DM ($\alpha = 5.5$)	4.59±0.05
Baseline + FM + MTD(CCT)	4.56±0.04
Baseline + FM +MTD(CCT) +DM ($\alpha = 5.0$)	4.57±0.05
Baseline + FM +MTD(CCT) +DM ($\alpha = 5.5$)	4.58±0.03
Baseline + FM +MTD(CCT) +DM ($\alpha = 5.6$)	4.58±0.04
Baseline + FM +MTD(CCT) +DM ($\alpha = 5.7$)	4.58±0.04
Baseline + FM +MTD(CCT) +DM ($\alpha = 6.0$)	4.57±0.04

Table 7. Results on Oxford from different α in DMs

Method	Oxford
Baseline +FM	3.86±0.08
Baseline +FM +MTD(CCT)	4.08±0.08
Baseline +FM +MTD(CCT) +DM($\alpha=5.5$)	4.09±0.07
Baseline +FM +MTD(CCT) +DM($\alpha=5.6$)	4.10±0.08
Baseline +FM +MTD(CCT) +DM($\alpha=5.7$)	4.08±0.09
Baseline +FM +MTD(CCT) +DM($\alpha=6.0$)	4.07±0.09

6. CONCLUSION

In this paper, we propose a novel ALG-GAN method for efficient text-to-image synthesis. Compared with previous models, our ALG-GAN performs better in generating consistent and high-quality images because the generator learns more adaptively with the forgetting mechanism and the drafting mechanism. Besides that, the comprehensive guiding discriminators reduces the mode collapse.

As future work, using efficient language model to process text description can obtain more informative condition vector, and using this vector to generate text to image can obtain higher quality images.

ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China under Grant No.62173265, the Fundamental Research Funds for the Central Universities, the Innovation Fund of Xidian University.

REFERENCE

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672-2680.
- [2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-

- consistent adversarial networks,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223-2232.
- [3] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” arXiv preprint arXiv:1805.08318, 2018.
 - [4] X. Chen, Y. Duan, R. Houthoof, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in Advances in neural information processing systems, 2016, pp. 2172-2180.
 - [5] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv preprint arXiv:1411.1784, 2014.
 - [6] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.
 - [7] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017, pp. 2642- 2651.
 - [8] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, “Generating images from captions with attention,” arXiv preprint arXiv:1511.02793, 2015.
 - [9] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, “Draw: A recurrent neural network for image generation,” arXiv preprint arXiv:1502.04623, 2015.
 - [10] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” arXiv preprint arXiv:1605.05396, 2016.
 - [11] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “Stackgan: Text to photorealistic image synthesis with stacked generative adversarial networks,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5907-5915.
 - [12] Zhang H, Xu T, Li H, et al. Stackgan++: Realistic image synthesis with stacked generative adversarial networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(8): 1947-1962.
 - [13] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, “Attngan: Fine-grained text to image generation with attentional generative adversarial networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp.1316-1324.
 - [14] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltechucsd birds-200-2011 dataset,” 2011.
 - [15] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. IEEE, 2008, pp. 722-729.
 - [16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in Advances in neural information processing systems, 2016, pp. 2234-2242.
 - [17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in Advances in Neural Information Processing Systems, 2017, pp. 6626-6637.
 - [18] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, “Tac-gan-text conditioned auxiliary classifier generative adversarial network,” arXiv preprint arXiv:1703.06412, 2017.
 - [19] M. Cha, Y. L. Gwon, and H. Kung, “Adversarial learning of semantic relevance in text to image synthesis,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 3272-3279.
 - [20] Z. Zhang, Y. Xie, and L. Yang, “Photographic text-to-image synthesis with a hierarchically-nested adversarial network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp.6199-6208.
 - [21] T. Qiao, J. Zhang, D. Xu, and D. Tao, “Mirrorgan: Learning text-to-image generation by redescription,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1505-1514.

AUTHORS

Yuting Xue was born in 1996. He received the M.S. degree in Electronic Science and Technology from Xidian University, Xi'an, China, in 2021. His main research interests include image and video processing, data mining.



Heng Zhou now is working toward the Ph.D. degree in Electronic Science and Technology with Xidian University, Xi'an, China. His current research interests include image processing, pattern recognition, and their applications in infrared target detection and segmentation.



Yuxuan Ding was born in 1995. He received the B.S. degree in Intelligent Science and Technology from Xidian University, Xi'an, China, in 2018. He is currently a Ph. D. Candidate at School of Electronic Engineering, Xidian University. His main research interest covers Machine Learning, Computer Vision, Vision-Language, and their applications.



Xiao Shan now is working toward the M.S. degree in Electronic Science and Technology with Xidian University, Xi'an, China. Her main research interests include image processing, machine learning, and image generation.



TRUST FOR BIG DATA USAGE IN CLOUD

Hazirah Bee Yusof Ali and Lili Marziana Abdullah

Kulliyyah of Information and Communication Technology (KICT),
International Islamic University Malaysia, Kuala Lumpur,
Malaysia

ABSTRACT

Integrating big data with an agile cloud platform can significantly affect how businesses achieve their objectives. Many companies are moving to the cloud, but the trust issue seemed to make a move to the cloud slower. This paper investigated the factors that affect Service Satisfaction that led to Trust. Since the sample was not normally distributed, the researchers used the PLS-SEM tool to analyse the relationship of the variables. The variables are Data Security, Data Privacy, Cloud Benefits, Reputation, Service Level Agreement (SLA), Risk Management, Service Satisfaction and Trust. The variables were linked together based on the analysis from qualitative research supported by theories, and the linkages were being validated through quantitative data analysis. The quantitative data analysis found that Data Security, Cloud Benefits, Reputation and SLA influence Service Satisfaction and Service Satisfaction influences trust.

KEYWORDS

Trust, Big Data, Cloud.

1. INTRODUCTION

Big data and cloud computing have altered the way companies operate. By the end of the day, most business and leisure activities are conducted in the cloud. Companies cannot afford not to maximise big data in the cloud because the benefits of the cloud are too appealing [1]. Similarly, several companies are there to provide solutions for analysing large datasets, assisting in disseminating helpful information to the general public [2]. Powerful data analysis tools, combined with the cloud's large storage capacity, enable businesses to understand their data better and, as a result, implement excellent solutions and improve their decision-making skills [3]. Cloud is a platform that provides solutions for increasingly complex businesses, including Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Business Intelligence (BI), and Document Management Systems (DMS), among others [3]. Cloud computing is becoming increasingly important because of the ever-increasing need for internet services and communications [2].

However, there are security concerns due to the high number of services, data, and devices housed in cloud environments [2]. Businesses are still reluctant to move sensitive data to the cloud, despite its many advantages and growing popularity [4]. Businesses are highly cautious about entrusting their most sensitive data to a cloud service provider because they lack confidence in the cloud service [5]. Storage data in the cloud means building Trust among cloud members, and having enough Trust is a critical challenge for the widespread adoption of big data in cloud computing [6][7]. As a result, the paper proposes organisations' Trust in big data and cloud computing using the organisation system-based theory [8].

This paper is divided into a few sections. Section 2 is the motivation of why the research was conducted. Here, the objective of the study is defined. Section 3 explains the methodology. Section 4 displays the data analysis process, and Section 5 explains descriptive statistics, while Section 6 presents the reflective measurement scale. After explaining the reflective measurement scale, Section 7 talks about the validation process, and Section 8 analyses data from the angle of the structural model. Section 9 illustrates the revised model along with results and discussions based on the data analysis, Section 10 states the future works, and Section 11 concludes the paper.

2. MOTIVATION

The investigation that contributes to Service Satisfaction leading to trust can be used as a guideline to ensure the successful usage of big data in the cloud [9][10]. Thus, the factors acquired from Qualitative Data Analysis supported by theories need to be validated.

The paper's objective is to present the findings of the quantitative data analysis on the factors that contribute to Service Satisfaction, which leads to Trust. The investigation through quantitative data analysis validates the framework captured from the analysis of qualitative research, which is supported by the Organization theory [11][12][13].

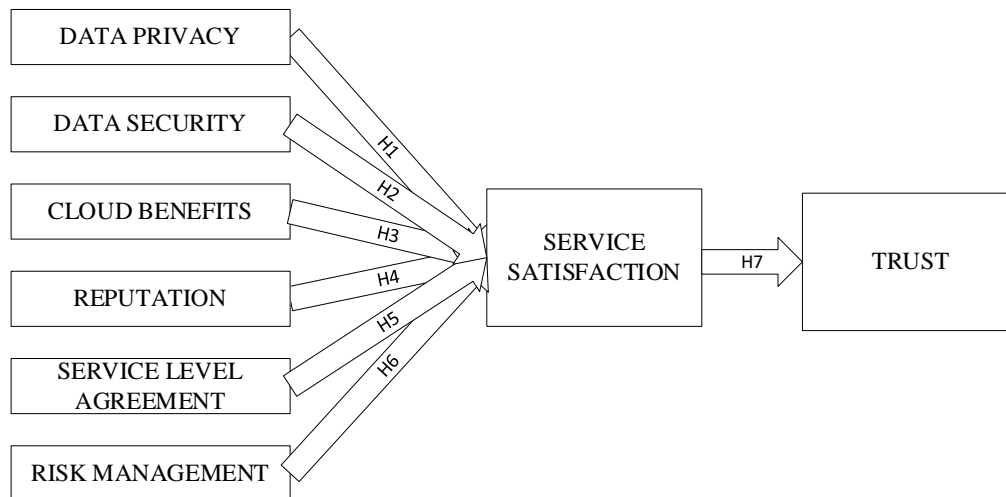


Figure 1. Organization System Theory: Trust for Big Data in Cloud

3. METHODOLOGY

Mixed method research is chosen where the qualitative method is done first, followed by a quantitative research method. By doing qualitative research first, the researchers are able to get more input from knowledgeable IT candidates. The interview questionnaire consists of open-ended questions, which enables the interview candidates to elaborate when answering the questions. The analysis of collected data supported by theories helps the researcher to produce a research framework.

The sample for this study was chosen through purposeful sampling. The researchers utilised purposeful sampling to discover IT personnel in charge of big data utilisation in the cloud and quickly and accurately answer big data and cloud questions.

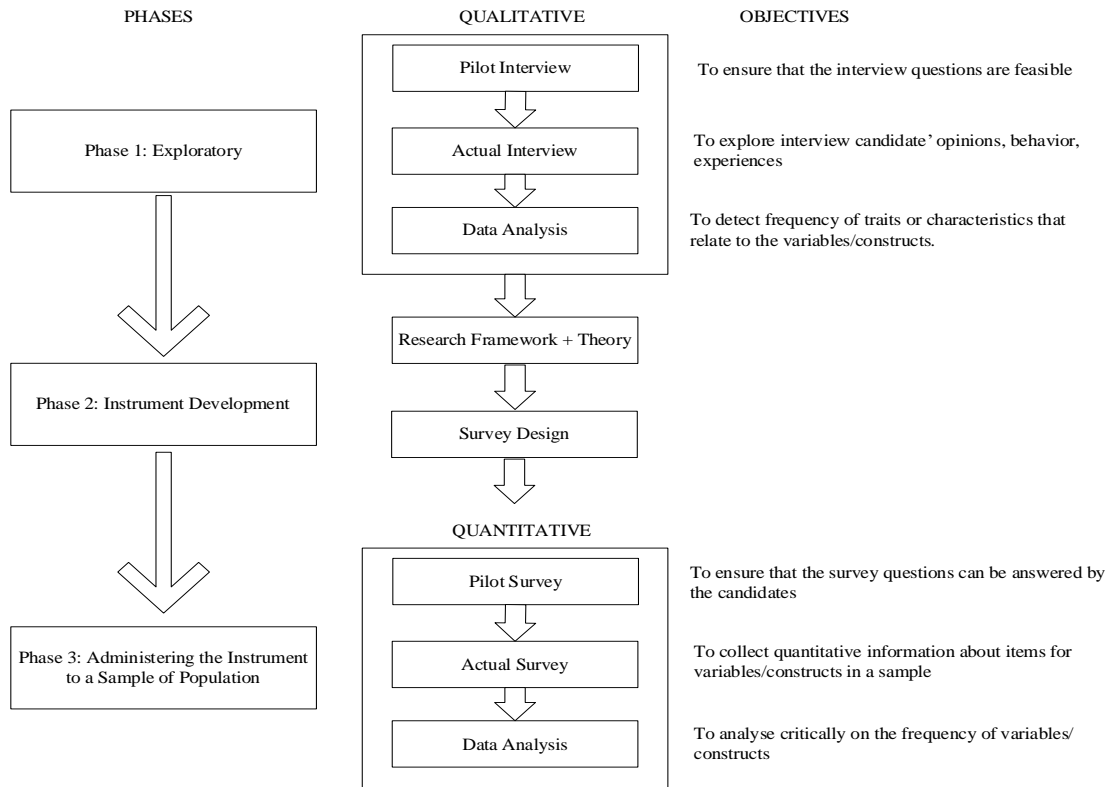


Figure 2: Methodology

Data collection was done during the pandemic when most people worked from home. So, the researchers used the online tool to communicate with them. The survey questionnaire was uploaded to Google Form in the cloud, and the link was given to prospective participants to complete the questionnaires. The participants were contacted via social media platforms such as WhatsApp and LinkedIn. The majority of the participants were acquired using LinkedIn.

Table 1: Quantitative Data Collection

Task	Start Date	End Date	No of Days
Pilot Data Collection	30 March 2021	6 April 2021	7 days
Actual Data Collection	7 April 2021	22 July 2021	107 days

Pilot data collection started on March 30th, 2021, and ended once data reached 30. The pilot data collection took seven days to complete. The researchers used the findings from pilot data analysis to confirm that the questions in the questionnaire are correct and related to the candidates so that they can be answered by the rest of the sample accurately.

Once data analysis for the pilot was completed, the actual data collection began. The actual data collection took 107 days, from 7 April 2021 to 22 July 2021. Linked-in enables the selection of candidates based on specific criteria so, the task of looking for suitable candidates became much more manageable. As a result, the researchers managed to get a sufficient number of candidates for purposeful sampling research.

To ensure suitable candidates, the researchers used three screening levels. The first screening is by using the permission letter to conduct the survey, and the second screening is on the front page of the questionnaire, where it states the objective of the survey briefly. Finally, the third screening is at Question 6 of the questionnaire, which says: "Does your organisation use cloud computing services?" If they select "No", then the questionnaire will exit. The researchers believe that the research output will be irrelevant if the candidates are chosen wrongly.

For this research, PLS-SEM is used to handle eight (8) constructs with many indicators for each construct. The indicators range from three (3) to six (6) indicators per construct.

4. DATA ANALYSIS PROCESS

The collected quantitative data was analysed based on the guidelines of Figure 3 where the researchers followed all the processes of descriptive statistics, measurement model, and structural model. As data analysis is done using PLS-SEM, the researchers started with descriptive statistics to investigate the extent of big data usage in the cloud for the organisation.

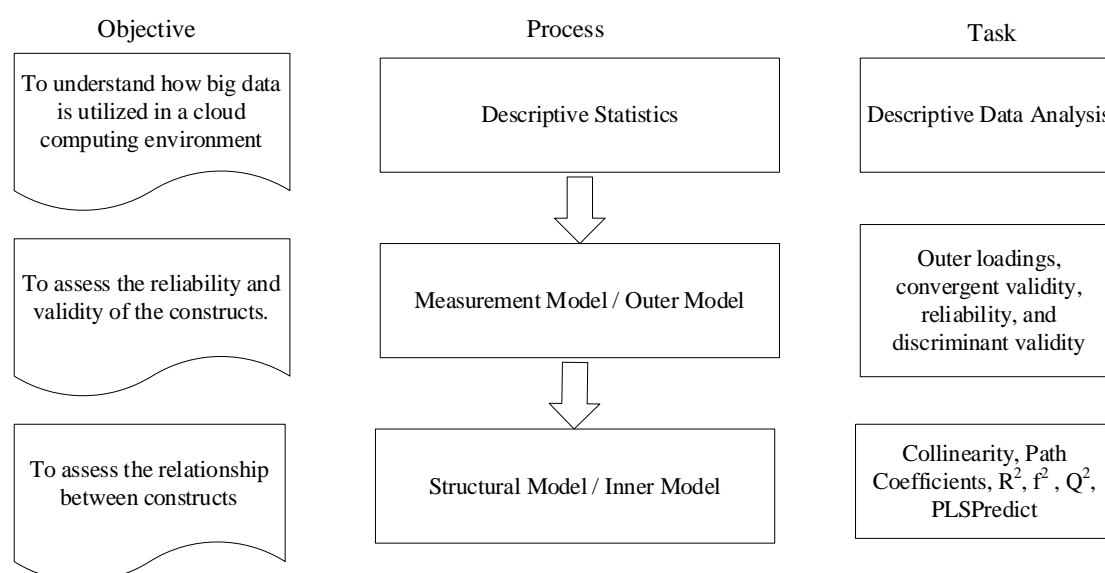


Figure 3: PLS SEM Data Analysis Process

Then the researchers continued with a measurement model or outer model to assess the reliability and validity of the construct. Finally, the structural model or inner model helps the researchers assess the relationship between constructs. Going through all the processes in Figure 3, the researchers determined which relationship of constructs are significant and insignificant.

5. PLS-SEM DESCRIPTIVE

SEM descriptive data analysis is the first process that the researchers implemented for the actual data analysis, as displayed in Table 2. The actual data collected was analysed descriptively so that the researchers could investigate the extent to which big data is utilised in a cloud computing environment.

Table 2: PLS-SEM Descriptive

Variable	Category	Quantity	Percentage(%)
Role	Director/CEO/Owner		14.6
	Manager		35.2
	Executive		32.4
	Others		17.8
Gender	Male		70.8
	Female		29.2
Age	Less than 21		1.4
	21 – 30		8.7
	31 – 40		28.3
	41 -50		37.4
	Above 50		24.2
Highest Education	Secondary		2.6
	College/Matriculation/Polytech		9.1
	University		87.7
Years with company	1-2 years		23.7
	3-5 years		21.5
	6-10 years		13.7
	More than 10 years		41.1
IT Knowledge	None at all		0
	Minimal		32.4
	Knowledgeable		56.6
	Very knowledgeable		32.4
Cloud Experience	1-2 years		25.7
	3-4 years		25.2
	More than 4 years		49.1
Organization Involvement	Services (IT-based)		54.5
	Services (Non-IT based)		21.3
	Others		24.2
Type of Organization	Sole proprietorship		17.1
	Sdn Bhd		24.4
	GLC Company		20.9
	Government		9.8
	Others		27.8
Years Established	1-5 years		11.9
	6-10 years		8.9
	11-15 years		13.7
	More than 15 years		65.5
Number of Employees	Less than 200		33.2
	More than 200		66.8
Number of IT Staffs	1-5		22.5
	6-10		8.1
	More than 10		69.4

The profile is suitable for the research since the target candidates are experienced enough in the management of big data and the cloud. Most of the time, experience tally with the length of service, position, age, and education. Being a male-dominated group indicates strong networking and collaboration among them in using cloud computing services.

A significant percentage of the sample consists of those with more than ten years of experience, know about IT technology and specifically cloud computing itself. Therefore, the sample is experienced and knowledgeable in big data and cloud, and that specific characteristic enabled them to answer the questionnaire comfortably.

The data shows that Sdn Bhd companies have the highest percentage with more than 50% of the organisation are more than 15% years of establishment, and most of them have more than 200 employees. This information tells us that the organisations are well established with many years of big data in cloud experiences.

The sample is using big data services regardless of structured, semi-structured and unstructured businesses.

More organisations prefer Microsoft Azure as their cloud platform than Google, Amazon and Cloudera, and Cloudera has the least number of organisations using it as the cloud platform.

In As a Service business model, they may choose more than one service business model. SaaS includes DBMS, CAD, MYOB, CRM, MIS, ERP, HRM, LMS, CM, and GIS. PaaS includes facilities for application design, application development, testing, and deployment. IaaS deals with virtual machines, storage, firewalls, load balancers, IP addresses, VLANs, and software bundles. DaaS involves cleaning, enriching data, and offering it to different systems, applications, or users. Finally, XaaS combines SaaS, PaaS, and IaaS offerings. So, we can conclude that organisations use more than one cloud component as a service.

Many organisations prefer a hybrid cloud as a delivery model. The choice of hybrid may be due to their need to use the cloud as a service but are hesitant to use the public cloud due to security and privacy issues. Because of that, they prefer hybrid where some parts of the cloud services are from the public cloud, and others remain in a private cloud. The hybrid cloud choice is to ensure that they are satisfied with the service and Trust to benefit from using big data in the cloud.

Half of the sample work with IT companies, and more than half are big organisations with more than 200 employees. 83.3% of the sample claimed to use structured data, 63.5% used semi-structured data, while others used unstructured data such as word, pdf, and social media. All of them are using cloud platforms such as Microsoft Azure, Google, Amazon, and Cloudera.

The selection of a proper sample is essential to investigate the factors that contribute to Service Satisfaction and finally towards Trust. The Trust in the usage of big data in the cloud is specifically targeted to those in charge of the big data usage for the organisation. If they do not trust, then their chances to use the cloud will be reduced. To summarise, the analysis of the descriptive statistics above does help the researchers investigate the extent of big data usage in the cloud. There seemed to be some issues in the usage of cloud services that relate to service satisfaction and Trust.

The indicators are taken from the itemised questions of the quantitative survey questionnaire. The indicators are labelled based on the variables where DP stands for Data Privacy, and DS stands for Data Security, CB for Cloud Benefits and so on. Quantitative data is analysed using PLS-

SEM Data Analysis. The researchers managed to get Figure 6 as the final outcome. Figure 6 is the adjusted research framework considering the loading, reliability, and validity of Figure 4.

6. REFLECTIVE MEASUREMENT MODEL

The model has a reflective measurement scale in which latent constructs (variables) whose indicators are influenced, affected, or caused by the latent variable [14]. All indicators will change when the latent construct changes. The indicators serve as empirical substitutes (proxy variables) for the latent variables [15]. Because the indicators are strongly connected and interchangeable, they must be rigorously tested for reliability and validity.

Data Privacy is made up of four (4) observed indicators: DP1, DP2, DP3, and DP4. Data Security is made up of three (3) observed indicators: DS1, DS2, and DS4. DS3 has been removed due to low factor loading. Cloud Benefits has five (5) indicators: CB1, CB2, CB3, CB4 and CB5. Reputation has three (3) indicators: REP1, REP2, and REP3. SLA has four (4) indicators. Risk Management has three (3) indicators. Both Service Satisfaction and Trust have six (6) indicators each.

Outer loadings, composite reliability, AVE, and square root should be examined and reported. In a reflective measurement scale, the causality direction goes from the blue-colour latent variable to the yellow-colour indicators of Figure 6.

7. VALIDATION PROCESS

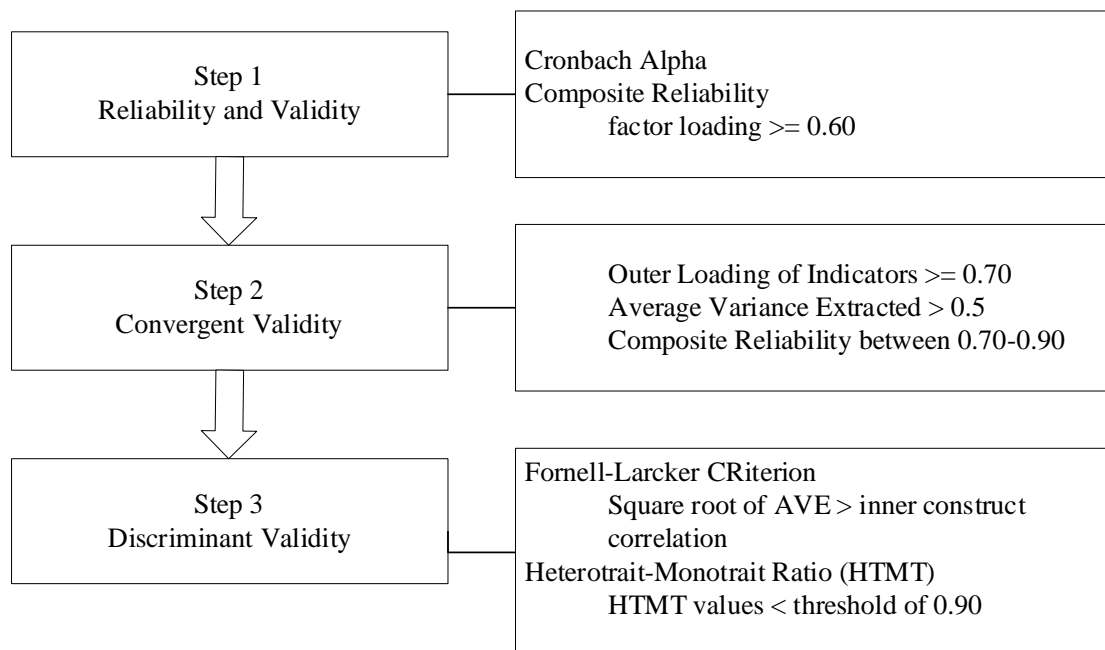


Figure 4: Validation Process

The measurement model /outer model is to assess the reliability and validity of the constructs. The relationship between the constructs and the indicators is shown in the measurement model, also known as the outer model. Outer loadings, convergent validity, reliability, and discriminant validity are used to assess the link between indicators and their constructs [14]. The PLS

Algorithm can be used to create a measurement model. [16] and [17] provide detailed explanations of how the basic PLS algorithm operates as implemented in SmartPLS 3.0 [14].

Table 3: Types of Variables and Indicators

Independent	Indicator	Mediator	Indicator	Dependent	Indicator
Data Privacy	DP1 DP2 DP3 DP4	Service Satisfaction	SS1	Trust	T1
			SS2		T2
			SS3		T3
			SS4		T4
Data Security	DS1 DS2 DS3 DS4		SS5		T5
			SS6		T6
Cloud Benefits	CB1 CB2 CB3 CB4 CB5				
Reputation	REP1 REP2 REP3				
Risk Management	RM1 RM2 RM3 RM4				

7.1. Step 1: Reliability and Validity

Table 4: Loading, Reliability and Validity

	Loading	Cronbach Alpha	Composite Reliability	Ave
CB1	0.744	0.870	0.906	0.658
CB2	0.807			
CB3	0.835			
CB4	0.843			
CB5	0.821			
DP1	0.703	0.792	0.865	0.617
DP2	0.880			
DP3	0.809			
DP4	0.738			
DS1	0.838	0.794	0.880	0.710
DS2	0.907			
DS4	0.777			
REP1	0.869	0.897	0.936	0.829
REP2	0.935			
REP3	0.925			
RM2	0.858	0.906	0.941	0.843

RM3	0.947			
RM4	0.947			
SLA1	0.824	0.859	0.904	0.703
SLA2	0.838			
SLA3	0.819			
SLA4	0.871			
SS1	0.864	0.937	0.950	0.762
SS2	0.886			
SS3	0.917			
SS4	0.909			
SS5	0.884			
SS6	0.770			
T1	0.807	0.909	0.930	0.688
T2	0.849			
T3	0.822			
T4	0.841			
T5	0.812			
T6	0.843			

The researchers tested the reliability and validity of the variables using Cronbach Alpha and Composite Reliability (CR). The indicators in Table 3 are labelled based on the variables in Figure 1 where DP stands for Data Privacy, DS stands for Data Security, CB for Cloud Benefits and so on.

Indicators with factor loading less than 0.6 were removed. Three (3) items (DS3, RM1 and RM5) were removed from the analysis because of low factor loadings (<0.600). The results for reliability and validity, along with the factor loadings for the items, are presented. The variables (constructs) are reliable and valid. Indicators DS1, DS2, and DS4 converge and measure Data Security. CB1, CB2, CB3, CB4, and CB5 converge and measure Cloud Benefits. The same applies to all other indicators in Table 3.

7.2. Step 2: Convergent Validity

Convergent validity refers to how indicators that measure the same construct agree with each other since they measure the same construct [16]. For convergent validity, the outer loadings of the indicators have to be at 0.70 or higher [18] average Variance extracted (AVE), $AVE > 0.50$ [16], and composite reliability (CR) should be between 0.70-0.90 [15]

All the Alpha values and CRs were higher than the recommended value of 0.700. The average Variance Extracted (AVE) and CRs were all higher or close to 0.5 and 0.7, which corroborates convergent validity.

Table 5: Heterotrait-Monotrait Ratio (HTMT)

	Cloud Benefits	Data Privacy	Data Security	Reputation	Risk Management	SLA	Service Satisfaction	Trust
Cloud Benefits	<i>0.811</i>							
Data Privacy	0.486	<i>0.786</i>						
Data Security	0.483	0.328	<i>0.842</i>					
Reputation	0.594	0.262	0.480	<i>0.910</i>				
Risk Management	0.489	0.412	0.449	0.447	<i>0.918</i>			
SLA	0.509	0.341	0.401	0.551	0.546	<i>0.838</i>		
Service Satisfaction	0.688	0.341	0.577	0.682	0.530	0.605	<i>0.873</i>	
Trust	0.663	0.365	0.546	0.604	0.551	0.677	0.830	<i>0.829</i>

Note: Values in italic represent the square-root of AVE.

Step 7.3: Discriminant Validity

Fornell-Larcker Criterion

Discriminant validity was assessed by fornell-larcker criterion. The square root of AVE for the construct was greater than inter construct correlation.

Heterotrait-Monotrait Ratio (HTMT)

Discriminant validity was also assessed by Heterotrait-monotrait ratio of correlation [16] with values below the threshold of 0.90, Since all the values are below 0.90. then, discriminant validity is established.

8. STRUCTURAL MODEL/ INNER MODEL

Structural model assesses the relationship between variables. When the measurement model assessment is satisfactory, the next step is evaluating the structural model of PLS-SEM results. Figure 5 has reflective constructs where it consists of one mediator variable (Service Satisfaction), six independent variables (Data Privacy, Data Security, Cloud Benefits, Reputation, SLA, and Risk Management), and one (1) dependent variable (Trust). One indicator (DS3) and two indicators (RM1 and RM5) were removed due to low factor loading (< 0.60).

8.1. Step 1: Evaluate structural model collinearity

VIF values can be examined, and if they are below 3.0, then there is no issue with multicollinearity [17]. Table 6 shows that Inner VIF for Cloud Benefits, Data Privacy, Data Security, Reputation, Risk Management, and SLA toward Service Satisfaction is below 3. So, there is no multicollinearity issue. The same applies to Inner VIF of Service Satisfaction towards Trust.

Table 6: VIF Values

Construct	Service Satisfaction	Trust
Cloud Benefits	2.668	
Data Privacy	1.680	
Data Security	1.759	
Reputation	2.338	
Risk Management	1.950	
SLA	2.132	
Service Satisfaction		1.000
Trust		

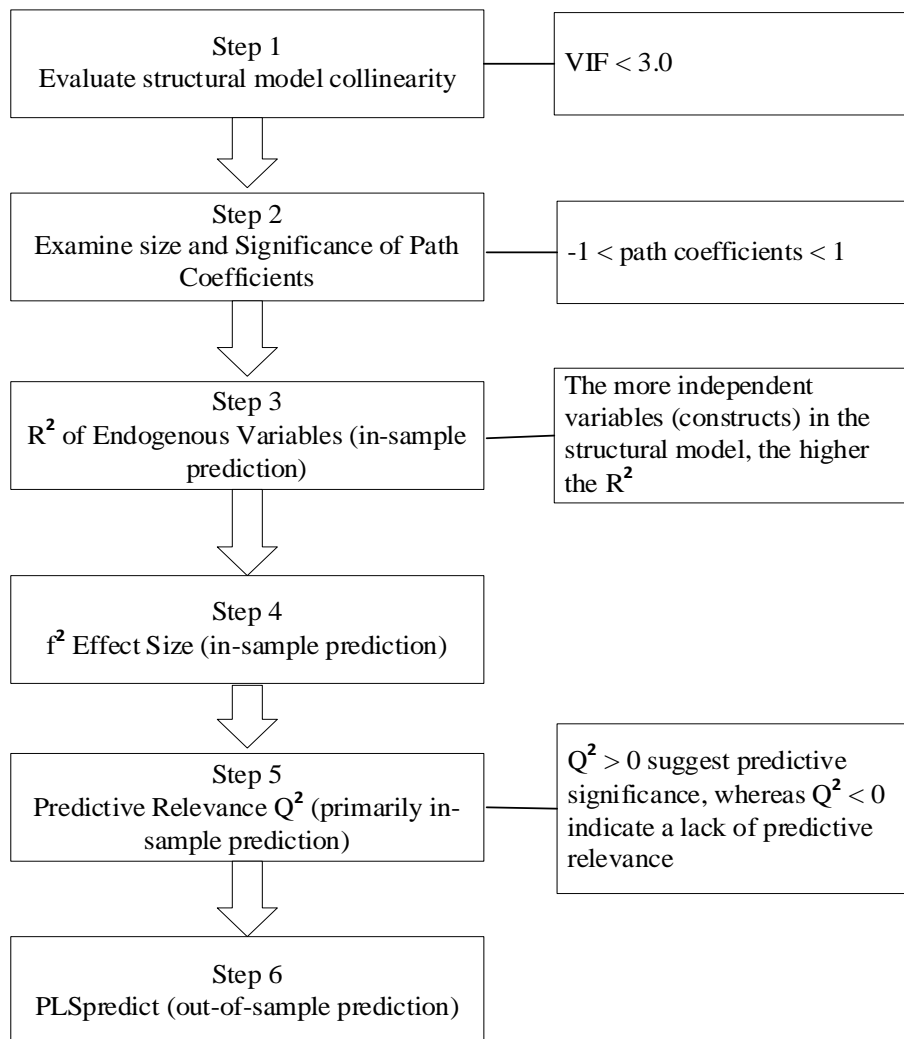


Figure 5. Structural Model

8.2. Step 2: Examine size and Significance of Path Coefficients

If multicollinearity is not an issue, the next step is to look at the path coefficients' size and significance. The path coefficients are standardised values ranging from +1 to 1 but rarely approaching +1 or 1, especially for complicated models when the structural model has numerous

independent constructs[15]. The researchers can use this method to test the hypothesised relationships among the constructs.

The path coefficient values are weaker in predicting dependent (endogenous) constructs the closer they are to 0. The stronger the dependent constructs, the closer they are to the absolute value of 1[15]. The researchers should test the structural model's prediction ability as the final step. The four metrics to analyse structural model prediction are outlined in steps 3 through 6 of Figure 5.

In Table 8, the path coefficient for Cloud Benefits is the highest at 0.383, followed by Reputation at 0.236, Data Security at 0.226, SLA at 0.201, Risk Management at 0.055, and finally, Data Privacy at -0.103. Path coefficient shows that Cloud Benefits are the strongest in predicting Service Satisfaction, followed by Reputation, Data Security, SLA, Risk Management, and Data Privacy. Data Privacy is the weakest in predicting Service Satisfaction. Service Satisfaction is strong in predicting Trust as the path coefficient of 0.896 is near +1.0.

Table 8: Significance of Path Coefficients

Construct	Service Satisfaction	Trust
Cloud Benefits	0.383	
Data Privacy	-0.103	
Data Security	0.226	
Reputation	0.236	
Risk Management	0.055	
SLA	0.201	
Service Satisfaction		0.896
Trust		

- The hypothesised path relationship between Cloud Benefits and Service Satisfaction is statistically significant.
- The hypothesised path relationship between Data Security and Service Satisfaction is statistically significant.
- The hypothesised path relationship between Reputation and Service Satisfaction is statistically significant.
- The hypothesised path relationship between SLA and Service Satisfaction is statistically significant.
- The hypothesised path relationship between Service Satisfaction and Trust is statistically significant.

However, the hypothesised path relationship between Risk Management and Service Satisfaction is statistically insignificant. The same applies to Data Privacy, where the hypothesised path relationship between Data Privacy and Service Satisfaction is statistically insignificant.

This is because the standardised path coefficients for Risk Management is 0.055 and for Data Privacy is -0.103, and both are significantly lower than 0.1. So, Risk Management and Data Privacy are poor predictors of Service Satisfaction and must be deleted. On the other hand, Cloud Benefits, Reputation, Data Security, and SLA are moderately strong predictors of Service Satisfaction, while Service Satisfaction is a significant predictor of Trust.

For path coefficient, in order to claim that the path is significant, weight of impact > 0.20 , t-value > 1.96 and P value < 0.05 . Path is insignificant if t value < 1.96 [15].

8.3. Step 3: R^2 of Endogenous Variables (in-sample prediction)

Data Privacy, Data Security, Cloud Benefits, Reputation, SLA, and Risk Management account for 73.8% of Service Satisfaction. Data Security, Cloud Benefits, Reputation, and Service Level Agreements (SLA) are significant, but Data Privacy and Risk Management are insignificant. Service Satisfaction is responsible for 80.5% of Trust.

Table 9: The coefficient of determination, R^2

	R^2
Service Satisfaction	0.739
Trust	0.803

The coefficient of determination, R^2 , is 0.803 for the Trust endogenous latent variable. The latent variables (Service Satisfaction) moderately explain 80.3% of the Variance in Trust.

R^2 is the most commonly used metric to assess structural model prediction in multiple regression models. R^2 is the coefficient of determination used to evaluate all endogenous constructs' in-sample prediction (Trust). The prediction is merely a measure of the predictive ability for the sample of data used in the calculations, and R^2 should not be inferred to the entire population [14]. R^2 is set to a minimum of 0.

The more independent variables (constructs) in the structural model, the higher the R^2 , the independent variables are related to the dependent variable constructs (Trust). Table 9 displays R^2 for Self-Satisfaction equal 0.739 and R^2 for Trust equal 0.803. For the Trust endogenous latent variable, the coefficient of determination, R^2 , is 0.803. This suggests that the latent variables (Service Satisfaction) account for 80.3% of the Variance in Trust. Data Privacy explains 73.9% of the Variance in Service Satisfaction, Data Security, Cloud Benefits, Reputation, SLA, and Risk Management.

8.4. Step 4: f^2 Effect Size (in-sample prediction)

R^2 , f^2 , and Q^2 predictive validity are useful in evaluating the predictive strength of a model based on in-sample data [15] In-sample prediction estimates the model. It predicts responses using the same sample, likely to exaggerate the model's predictive power. This is known as an overfitting problem (a greater forecast than is reasonable). It implies that the model may have limited value in predicting observations outside the original sample. When utilising PLS-SEM, [19] provided a method for assessing out-of-sample prediction. The technique entails first estimating the model on a training (analytical) sample and then using the model's outputs to predict other data in a separate holdout sample.

Table 10: f^2 effect size

Construct	Service Satisfaction	Trust
Cloud Benefits	0.210	
Data Privacy	0.024	
Data Security	0.111	
Reputation	0.091	
Risk Management	0.006	
SLA	0.073	
Service Satisfaction		4.086
Trust		

8.5. Step 5: Predictive Relevance Q^2 (primarily in-sample prediction)

Prediction is also assessed via the Q^2 value (blindfolding) [20]. When interpreting Q^2 , numbers greater than zero suggest predictive significance, whereas values less than zero indicate a lack of predictive relevance. Furthermore, Q^2 values greater than 0.25 and 0.50 show the PLS-SEM model's medium and sizeable predictive relevance, respectively.

Table 11 shows Q^2 for Service Satisfaction as 0.462 and Q^2 for Trust as 0.438. So we can conclude that both Service Satisfaction and Trust represent a medium predictive because both of them is higher than 0.2 but lesser than 0.5.

Table 11: Q^2

	Q^2
Service Satisfaction	0.462
Trust	0.438

8.6. Step 6: PLSpredict (out-of-sample prediction)

The RMSE should be used as the prediction statistic in most cases. However, the MAE should be used if the prediction error distribution is extremely non-symmetrical [19]. The RMSE values are compared to a naive value derived by a linear regression model to measure the prediction error of a PLS-SEM analysis (LM). This is done to make predictions for the measured variables (indicators). In the PLS path model, the LM process uses a linear regression model to predict each endogenous construct's indicators from all exogenous latent variable indicators. However, the stated model structure represented by the measurement and structural theory is not included in the LM process[19]. Depending on the symmetry of the prediction error distribution, the RMSE and MAE values are both acceptable prediction benchmarks.

- The model lacks predictive power when the RMSE or MAE has higher prediction errors than the naive LM benchmark for all dependent variable indicators.
- The model has low predictive power when the dependent construct indicators have higher prediction errors than the naive LM benchmark.
- The model has medium predictive power when an equal or minor number of dependent construct indicators have higher prediction errors than the naive LM benchmark.
- The model has high predictive power when none of the dependent construct indicators has higher RMSE or MAE prediction errors than the naive LM benchmark.

Table 12: PLSpredict

	RMSE (PLS)	RMSE (LM)	MAE (PLS)	MAE (LM)
SS2	0.593	0.626	0.441	0.443
SS3	0.595	0.642	0.404	0.442
SS5	0.665	0.718	0.476	0.513
SS6	0.876	0.991	0.646	0.665
SS1	0.651	0.681	0.453	0.444
SS4	0.710	0.745	0.477	0.493
T6	0.653	0.660	0.431	0.416
T1	0.669	0.651	0.461	0.451

T4	0.811	0.843	0.629	0.637
T3	0.702	0.738	0.538	0.554
T2	0.753	0.811	0.557	0.575
T5	0.661	0.733	0.445	0.457

Table 12 indicates that the RMSE or MAE for PLS has lower prediction errors than the naive LM benchmark for all dependent variable indicators except T1 and T6, meaning that the model has predictive power. Compared to the naive LM benchmark, T1 and T6 have MAE for PLS, and T1 has RMSE with more significant prediction errors.

The model has medium to high predictive power because most dependent construct indicators have lower prediction errors than the naive LM benchmark. From the quantitative data analysis above, we can conclude that Quantitative Data Analysis help the researchers to validate the research framework.

9. RESULT AND DISCUSSION

First, the descriptive statistics analysis helps the researchers investigate the extent of big data usage in a cloud computing environment. Most of them use big data extensively, yet they do not fully utilise cloud computing services since many prefer hybrid cloud compared to the public cloud. The findings from descriptive statistics analysis can help the researchers answer some of the research questions.

Second, Figure 6 is produced after considering the reliability and validity of the constructs. Some indicators are removed (DS3, RM1 and RM5) as the factor loading is less than 0.60. Thus, the measurement model helps assess the constructs' reliability and validity.

Finally, the structural model helps to assess the relationship between constructs. Following the guidelines from Step1 to Step 6 of the structural model in Figure 5, the researchers can conclude which relationship of constructs is significant and insignificant. So, it is found that Data Security, Cloud Benefits, Reputation, and SLA are significant while Data Privacy and Risk Management are insignificant. Therefore, the findings validate the research framework.

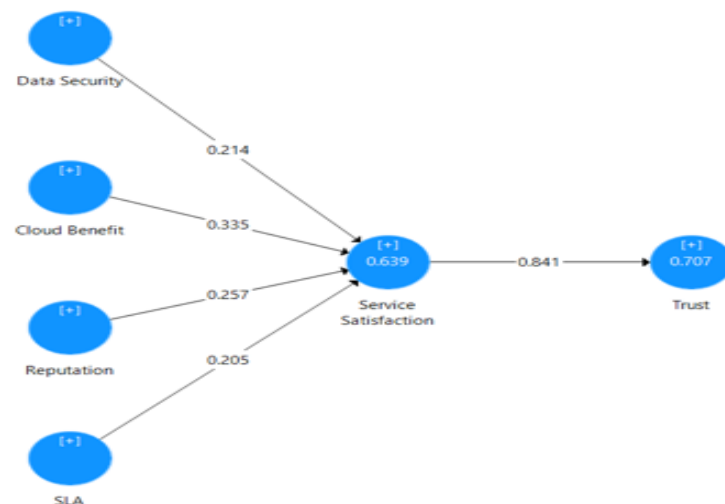


Figure 6: Revised Model

Finally, structural model helps to assess the relationship between constructs. Following the guidelines from Step1 to Step 6 of the structural model in Figure 5, the researchers can conclude which relationship of constructs is significant and insignificant. So, it is found that Data Security, Cloud Benefits, Reputation, and SLA are significant while Data Privacy and Risk Management are insignificant. Therefore, the findings validate the research framework.

Table 13: Hypothesis testing

H	Hypothesis	Supported/Not supported
H1	There is a significant positive relationship between data privacy and service satisfaction.	Not supported
H2	There is a significant positive relationship between data security and service satisfaction.	Supported
H3	There is a significant positive relationship between cloud benefits and service satisfaction.	Supported
H4	There is a significant positive relationship between reputation and service satisfaction.	Supported
H5	There is a significant positive relationship between service level agreement and service satisfaction.	Supported
H6	There is a significant positive relationship between risk management and service satisfaction.	Not supported
H7	There is a significant positive relationship between service satisfaction and Trust.	Supported

10. FUTURE WORK

The researchers recommend for the sampling be done thoroughly and accurately. Due to the differences in job scope in managing IT in the organisation, many IT managers perceived cloud usage from various angles. Some perceived the cloud as a place for big data storage, some perceived the cloud for data analytic purposes, while others may see the cloud as a platform to utilise. SaaS, PaaS and IaaS. So, purposive sampling should be done accurately and adequately. The sample needs to be screened thoroughly before they are being selected for the interview questionnaire. They must be the right candidate with the right position of IT in managing cloud usage. The screening is to ensure; they answer the questionnaire accurately and are able to represent the population. Selecting the right candidate can be difficult, but it does help in getting the most relevant constructs to represent the population.

11. CONCLUSIONS

This research is highly needed to benefit the cloud without any hesitation and unnecessary worries. Cloud users need guidelines on the criteria required before they decide to use the cloud for their big data processing, and to follow the guideline make them ready for all the uncertainties the cloud might have.

In conclusion, the research contributes to knowledge as it can give organisations guidelines on how to put their big data in the cloud with less worry. The benefits of the cloud far exceed the fear that they have. Because of the overwhelming benefits, cloud users transfer their anxiety on Data Security to the cloud providers using Reputation and SLA. The cloud providers must have an excellent image so that cloud users can trust them. Besides Reputation, the Trust can be strengthened by having SLA with them. SLA enables the cloud users to transfer the responsibility of Data Security to the cloud providers.

ACKNOWLEDGEMENTS

This research has been funded by Fundamental Research Grant Scheme (FRGS16-022-0521) supported by Ministry of Higher Education, Malaysia.

REFERENCES

- [1] B. M. Balachandran and S. Prasad, "Challenges and Benefits of Deploying Big Data Analytics in the Cloud for Business Intelligence," *Procedia Comput. Sci.*, vol. 112, pp. 1112–1122, 2017.
- [2] A. Sether, "Cloud Computing Benefits," *SSRN Electron. J.*, no. January, 2016.
- [3] K. Tatic, "The benefits of using cloud technology in Bosnia and Herzegovina," no. April, 2020.
- [4] S. M. Din, R. Ramli, and A. A. Bakar, "A Review on Trust Factors affecting Purchase Intention on Instagram," *2018 IEEE Conf. Appl. Inf. Netw. Secur.*, no. November, pp. 49–53, 2018.
- [5] F. Selnes, "Consequences of Trust and Relationships," *Eur. J. Mark.*, vol. 32, no. 3/4, pp. 305–322, 1998.
- [6] J. K. Adjei, "Explaining the role of trust in cloud computing services," *Info*, vol. 17, no. 1, pp. 54–67, 2015.
- [7] Z. Tbatou, A. Asimi, and C. El Balmany, *Trust in Cloud Computing Challenges: A Recent Survey*, vol. 81, no. January. Springer International Publishing, 2020.
- [8] R. R. Greene, *General systems theory*, no. January 2013. 2017.
- [9] F. Zohra Filali and B. Yagoubi, "Global Trust: A Trust Model for Cloud Service Selection," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 5, pp. 41–50, 2015.
- [10] M. Felici and S. Pearson, "Accountability, Risk, and Trust in Cloud Services: Towards an Accountability-Based Approach to Risk and Trust Governance," *2014 IEEE World Congr. Serv.*, pp. 105–112, 2014.
- [11] I. Encalada, "General systems theory . Practical approach in education," no. October, 2019.
- [12] A. Šijan, D. Karabašević, and D. Rajčević, "The importance of the general system theory for the modern world," *Trendovi u Posl.*, vol. 7, no. 2, pp. 87–94, 2019.
- [13] A. Steblyanskaya, Y. Jun, and M. Vasiev, "General systems theory for sustainable green development," no. September, pp. 138–141, 2021.
- [14] J. F. Hair, C. M. Ringle, and M. Sarstedt, "PLS-SEM : Indeed a Silver Bullet," vol. 19, no. 2, pp. 139–151, 2011.
- [15] J. F. Hair, J. J. Risher, M. Sarstedt, and C. M. Ringle, "When to use and how to report the results of PLS-SEM," *Eur. Bus. Rev.*, vol. 31, no. 1, pp. 2–24, 2019.
- [16] J. Risher, "When to use and how to report the results of PLS-SEM," no. December, 2018.
- [17] J. F. Hair, M. C. Howard, and C. Nitzl, "Assessing measurement model quality in PLS-SEM using confirmatory composite analysis," *J. Bus. Res.*, vol. 109, no. December 2019, pp. 101–110, 2020.
- [18] M. Sarstedt, C. M. Ringle, and J. F. Hair, *Handbook of Market Research*, no. September. 2020.
- [19] P. N. Sharma, M. Sarstedt, G. Shmueli, K. H. Kim, and K. O. Thiele, "PLS-Based Model Selection: The Role of Alternative Explanations in Information Systems Research," *J. Assoc. Inf. Syst.*, no. April, pp. 346–397, 2019.
- [20] F. Ali, S. M. Rasoolimanesh, and C. Cobanoglu, "Applying Partial Least Squares in Tourism and Hospitality Research," *Appl. Partial Least Squares Tour. Hosp. Res.*, no. September, pp. 1–264, 2018.

AUTHORS

The Author is a researcher and also a lecturer at UNIKL Business School Kampung Baru. She is a PhD candidate in Kulliyyah of Information and Communication Technology (KICT), IIUM and in the process to submit her thesis.



The Co-Author is currently an Associate Professor at the Department of Information Systems, Kulliyyah of Information and Communication Technology (KICT), IIUM.



© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

USING DOMAIN KNOWLEDGE FOR LOW RESOURCE NAMED ENTITY RECOGNITION

Yuan Shi

School of Computer Science & Technology,
Beijing Institute of Technology, Beijing, China

ABSTRACT

In recent years, named entity recognition has always been a popular research in the field of natural language processing, while traditional deep learning methods require a large amount of labeled data for model training, which makes them not suitable for areas where labeling resources are scarce. In addition, the existing cross-domain knowledge transfer methods need to adjust the entity labels for different fields, so as to increase the training cost.

To solve these problems, enlightened by a processing method of Chinese named entity recognition, we propose to use domain knowledge to improve the performance of named entity recognition in areas with low resources. The domain knowledge mainly applied by us is domain dictionary and domain labeled data. We use dictionary information for each word to strengthen its word embedding and domain labeled data to reinforce the recognition effect. The proposed model avoids large-scale data adjustments in different domains while handling named entities recognition with low resources. Experiments demonstrate the effectiveness of our method, which has achieved impressive results on the data set in the field of scientific and technological equipment, and the F1 score has been significantly improved compared with many other baseline methods.

KEYWORDS

Named Entity Recognition, Domain Knowledge, Low Resource, Domain Dictionary.

1. INTRODUCTION

Named Entity Recognition (NER), also known as entity extraction technology, is one of the key technologies of natural language processing. Its goal is to identify the predefined entity categories from unstructured text, such as person, location, organization, etc. NER is conventionally formulated as a sequence labeling problem, mainstream method which has been widely used is neural network model^[1] and pretrained Language Models (LMs)^[2], for example, we often use Long Short Term Memory(LSTM)^[3] or Bidirectional Encoder Representation from Transformers(BERT)^[4] as encoding layer, multi-layer perceptron plus Softmax^[5] or Conditional Random Field (CRF)^[6] as decoding layer.

Supervised neural network model usually needs to use a large amount of labeled data for training. However, except for some mature fields, such as newswire domain, most other fields lack sufficient high-quality annotated corpus. When the annotated corpus is small, the existing method of entity recognition using deep neural network will have significant performance degradation. Due to the lack of sufficient training data, these models are unable to fully learn the implicit feature representation. It can be seen that named entity recognition in the field of resource scarcity is more difficult than that in the mature field.

In recent years, in order to solve the problem of named entity recognition in the field of resource scarcity, many methods have been proposed. Li used a large parallel corpus to project information from high resource domain to low resource domain^[7]. Yang use cross-resource word vector to bridge low resource domain and high resource domain to realize knowledge transfer^[8]. Chen and Ding propose explicitly connecting entity mentions based on both global coreference relations and local dependency relations for building better entity mention representations^[9], which is desirable for domain-specific NER.

In our opinion, the problem of cross-domain knowledge transfer is that there are different entity categories in different fields. For example, in the newswire domain, we usually identify persons and locations, while in the science and technology equipment domain, we need to identify products and achievements. Since the decoding layer needs to be trained and tested with a consistent set of tags, the need to adjust the output layer to retrain in the new field may further increase training costs.

Based on this situation, we propose the use of domain knowledge to improve the problem of named entity recognition in resource scarcity domains. Specifically, analogy to the method of using word embedding to strengthen character embedding in Chinese NER, we introduce domain dictionaries and small-batch domain labeling data into this scenario, which effectively alleviates the NER problem in low resource domains. The feasibility of adopting this method is that although there is a lack of mature annotation data in many fields of scarce resources, some experience-generated domain dictionaries with rich information can be applied. These domain dictionaries contain a large number of entity information, which can effectively improve the recognition rate of domain-specific words. Our method is to scan the input sentence, for every word, find out whether it is located in a phrase in the domain dictionary. The dictionary vector is obtained by using the formula defined in subsequent chapters, and then spliced with pre-trained word embedding to obtain the enhanced word embedding. Small-batch domain labeling data are generated by concentrating a small number of professionals in a short period of time, we use it to replace generic domain data used by existing models as training set to achieve better recognition results. In this way, when dealing with different fields, only the dictionary and training data in this field need to be replaced. The structure of existing model itself needs no modification, which saves a lot of training costs.

In section 2, we introduce you some technical background. In section 3, we describe the principle and implementation of our model in detail, we conduct our experiments on the low resource domain dataset in section 4, respectively. The results show that our method has achieved very competitive results on the resource-scarce domain dataset of scientific and technological equipment, and the F1 score is significantly improved compared with many baseline methods. Finally, we summarize our article in section 5.

2. BACKGROUND

Generally speaking, Named Entity Recognition (NER) is to extract entities from unstructured texts. Academically, named entities generally include three categories (entities, time expressions and numerical expressions) and seven subcategories (persons, locations, organizations, time, date, currency, and percentage). In specific work, more categories of entities can be defined according to requirements. The most common modeling method is to model named entity recognition as a sequence labeling problem, that is, for an input token sequence, after model processing, the corresponding label will be assigned to each token as the output.

Domain dictionaries are lists of named entities collected from various sources, such as disease name dictionary in the medical field and actor name dictionary in the film field. They have been

widely used in named entity recognition. Domain dictionaries are generally used to create features for the model, usually binary features, to indicate whether the corresponding n-gram exists in the dictionary. In addition, grammar features can be extracted from it, such as prefix and suffix, for rule matching.

Domain labeled data are necessary in the identification of named entities using neural network models. It will guide the model to learn implicit feature representation as training data, so that it can obtain the ability of identifying specific entities. In practical applications, due to the lack of mature annotation data in the field of resource scarcity, most of the existing methods use high-quality annotation corpus of general fields (such as newswire domain) as training dataset.

3. USING DOMAIN KNOWLEDGE

The basic model of our paper is LM-LSTM-CRF, namely, the ‘Language Model–Long Short-Term Memory–Conditional Random Field’ model. Based on this model, domain knowledge is added as an auxiliary to improve the effect of the model in identifying named entities in the field of resource scarcity.

3.1. LM-LSTM-CRF

LM-LSTM-CRF was proposed by Liu et al., 2018^[10], its main purpose is to solve the problem of lacking annotated corpus for model training in sequence annotation tasks. This model can be used for part-of-speech tagging, named entity recognition, noun phrase segmentation and other tasks that can be modeled as sequence annotation problems.

The model extracts knowledge from original text and assigns weight to the sequence annotation task. In pre-trained word vector, in addition to word level knowledge, character-aware neural language model is also introduced to extract character level knowledge. According to the experiment, when the tasks are inconsistent, if simply set them in a common training environment, the language model may affect the effect of sequence labeling. Therefore, the highway^[11] layer has been used to convert the output of character level to different semantic spaces to mediate and unify these two tasks, and guide the language model to master the knowledge of specific tasks. Different from most transfer learning methods, this framework does not rely on any other supervision, but extracts knowledge from the self-contained order information of the training sequence. Compared with previous methods, this task-specific knowledge can be trained more effectively using simpler models. A large number of experiments on the benchmark data set has been conducted to prove the effectiveness of using character-level knowledge and collaborative training in sequence annotation tasks.

Overall, this model uses the character-level bidirectional LSTM structure to implement the language model, and forms a multi-task transfer learning layer with the sequence annotation feature extraction layer implemented by the word-level bidirectional LSTM structure. Finally, the CRF layer is used for sequence decoding.

3.2. Incorporating Domain Knowledge into Model

3.2.1. Lattice-LSTM and Its Strengthening

In order to achieve the goal of integrating domain dictionary information into existing model, the method we apply is derived from Peng and Ma^[12]. The main purpose of their work is to improve the Lattice-LSTM model.

The Lattice-LSTM model was proposed by Zhang and Yang^[13]. For Chinese named entity recognition tasks, the model encodes input characters and all potential word sequences matched with dictionaries. Compared with previous character-based methods, this model can make better use of word and word sequence information. Compared with word-based methods, it can avoid named entity recognition errors caused by word segmentation errors. At the same time, due to the use of gated recurrent unit, the model can select the most relevant words from sentences to obtain better named entity recognition results.

Although the model has achieved some success in the task of Chinese named entity recognition, it still has a problem that cannot be ignored, that is, the model architecture is too complex. In essence, the model converts the input form of a sentence from a common chain sequence to a graph, which increases the computational cost of sentence modeling. It is this problem that limits its further application in the industrial field that requires real-time response.

Through the introduction of Lattice-LSTM, it can be seen that the advantage of Lattice-LSTM is to retain all possible matching words in the dictionary for each character, which can avoid the exploratory selection of character matching results and introduce error propagation. Therefore, if we want to optimize the model, we should retain this advantage as much as possible, meanwhile, try to overcome the problem of excessive modeling and computing costs caused by the conversion of input forms. In response to this, Peng and Ma proposed an improved method in [12].

The first method is based on Softword technology, which was originally used to merge word segmentation information into downstream tasks (Zhao and Kit, 2008^[14]. Peng and Dredze, 2016^[15]), mainly by embedding the corresponding segment labels to enhance character representation:

$$X_j^c \leftarrow [X_j^c; e^{seg}(seg(c_j))]. \quad (1)$$

Here X_j^c represents the character level vector, and $seg(c_j) \in y_{seg}$ denotes the segmentation label of the character c_j predicted by the word splitter, e^{seg} denotes the segmentation label embedding lookup table. Usually, $y_{seg} = \{B, M, E, S\}$, where B, M, E denote that the character is the beginning, middle, end of the word, and S denotes that the character itself constitutes a single word.

Based on this technology, a dictionary has been proposed to construct a word splitter, which allows a character to have multiple segmentation labels at the same time:

$$X_j^c \leftarrow [X_j^c; e^{seg}(segs(s)_j)]. \quad (2)$$

Here $e^{seg}(segs(s)_j)$ is a 5-dimensional binary vector, and each dimension corresponds to each item of {B, M, E, S, O}. This method is called ExSoftword.

On the surface, ExSoftword has successfully introduced dictionary information into vector representation. However, through analysis, it can be found that although ExSoftword attempts to save all dictionary matching results by allowing a single character to have multiple segmented labels, it still loses a lot of information, so that in many cases it cannot recover the matching results from the segmentation label sequence.

In order to solve this problem, not only the possible segmented label characters, but also the corresponding matching words should be retained. The specific approach is to correspond each character c of a sentence s to four word sets marked by four segmentation labels 'BMES'. The word set $B(c)$ consists of all dictionary matches beginning with c on S . Similarly, $M(c)$ consists of all lexicon matched words in the middle of which c occurs, $E(c)$ is composed of all dictionary matching words ending with c , and $S(c)$ is a word composed of c . At this point, if the word set is empty, add a special word 'NONE'. Next, compress the four word sets of each character into a fixed dimension vector. To retain information as much as possible, four word sets are concatenated as a whole and added to the character representation:

$$\begin{aligned} e^s(B, M, E, S) &= [v^s(B) \oplus v^s(M) \oplus v^s(E) \oplus v^s(S)], \\ X^c &\leftarrow [X^c; e^s(B, M, E, S)]. \end{aligned} \quad (3)$$

Here, v^s represents the function of mapping a single word set to a dense vector. This also means that each word set should be mapped to a fixed dimension vector. In order to achieve this goal, after trying the less successful mean-pooling algorithm, the word frequency has been used to represent its weight. Specifically, w_c is used to represent the character sequence of w , and $z(w)$ is used to represent the frequency of w_c in the statistical data set. Note that if w_c is covered by another word in the dictionary, the frequency of w is not counted. Finally, the weighted representation of the word set S is obtained by:

$$v^s(S) = \frac{1}{Z} \sum_{w \in S} (z(w) + c) e^w(w), \quad (4)$$

Where

$$Z = \sum_{w \in B \cup M \cup E \cup S} z(w) + c.$$

Here, the weights of all words in the four word sets are normalized to allow them compete with each other across sets. e^w indicates that the words are embedded in the query table. The addition of constant c is to introduce smoothing treatment to the weight of each word to increase the weight of rare words. Here, the value of c is set as: in the statistical data set, 10% of the training words appear less than c times.

To sum up, the word vector representation combined with dictionary information mainly includes the following four steps. First, a dictionary is used to scan each input sentence, and four word sets of 'BMES' are obtained for each character in the sentence. Second, query the frequency of each word appearing on the statistics set. Third, get the vector representation of four word sets for each character by formula, and add it to the character representation. Finally, based on the enhanced character representation, any appropriate neural sequence labeling model can be used for sequence labeling.

3.2.2. Using Domain Dictionary

In our experiment, the experimental data we use is English text from science and technology equipment domain, however, the word vector representation combined with dictionary information introduced above is carried out on Chinese texts. If we want to apply this method in our own model, we need to redesign it to realize the conversion from Chinese processing to English processing.

Due to the language differences between Chinese and English, there is some distinction on the definition of character level in the two languages. When dealing with Chinese, Chinese characters are used as the basic character units. Depending on the difference of word segmentation, words composed of single and multiple characters can be used as word level units. Relatively speaking, in English, letters are used as basic character units and words are used as word level units. The main reason for this is that the characters in Chinese have the ideographic function, while in English only words have practical meaning, a single letter has no meaning. The method mentioned above is implemented for Chinese. The existing dictionary is used for matching to enhance the representation of character-level vector (for Chinese, that is, a single word). When dealing with English text, the existing English dictionary in the field of science and technology equipment (including a single word or phrases of multiple words) has been used by us to enhance the representation of word-level vector.

According to the differences between English and Chinese, our main processing flow is as follows:

(1). Input English sentence S . For each word X , find out whether it is located in a phrase in the domain dictionary, and process all the matching results into four word sets corresponding to BMES. These word sequences contain all the word sequences of the current word. For example, for the word 'rocket', if there are 'rocket motor', 'multistage rocket' and 'meteorological rocket' in the dictionary, then $B(\text{rocket}) = \{\{\text{rocket motor}\}\}$, $M(\text{rocket}) = \{\text{None}\}$, $E(\text{rocket}) = \{\{\text{multistage rocket}\}, \{\text{meteorological rocket}\}\}$, $S(\text{rocket}) = \{\text{None}\}$.

(2). For the further apply, the word sequence set obtained in the previous step need to be compressed into a fixed-dimension vector, as shown in Formula (3), v^s is the function of mapping the word set to a dense vector. Its definition is shown in Formula (4), $z(w)$ uses the frequency of the word sequence to represent the weight. We combine training and testing data of the task to construct a statistical data set, and conduct frequency statistics on it. Finally, the fixed-dimension vector is obtained through Formula (4).

(3). We add obtained vectors into the word representation. In the existing LM-LSTM-CRF model, the input used at the word level is Glove 100-dimension data, and the pre-trained word embedding is fine-tuned. In order to make use of the domain dictionary information, we use Glove 100-dimension data and the word vector augmented by domain dictionary information as input at the word level.

3.2.3. Adding Labeled Data

In the field of resource scarcity, in addition to the dictionary, there are some unlabeled data that can be used. These data belong to the original corpus, which can be used as testing data. We annotate some original corpus of science and technology equipment domain, and use these labeled data to replace the CoNLL-2003^[16] set as a training set to retrain the model. The results show that higher F1 score is obtained by using the new model for experiments.

The text form of the original corpus is as follows:

*Launching the sixth branch of the US armed forces Defense News
Space
Launching the sixth branch of the US armed forces
By: Mike Rogers*

Vandenberg Air Force Base supported the successful launch of the fourth Iridium mission on a SpaceX Falcon 9 rocket on Dec. 22, 2017. (Tech. Sgt. Jim Araos/U.S. Air Force)

The labeling method used in our work is BIO tagging, and the final labeling result is a combination of two types of labels. The first type of label indicates that the current entity belongs to the category defined from the meaning of entity itself, entity categories defined in this annotation are PER (persons), ORG (organizations), PCT (products), OUT (Outcomes), SER (Services), TIM (time). The second type of label indicates the position of the current word in the named entity, the standard practice is to use BIO labeling, that is, each word is labeled as 'B-X', 'I-X' or 'O', and the 'X' is the first type of label indicating that the current word belongs to type X, and these labels indicate that the current word is located at the beginning, the middle or end position of the named entity and does not belong to any named entity.

As for the annotation management, we use the method of cross-examination annotation by multiple annotators, that is, each annotator independently labels the data of his assigned part, and each document will be labeled by at least three annotators. Then each annotator reviews the work of other annotators to form a review chain, which promotes the communication between annotators and the unity of data understanding. The final annotation results of the data are obtained by voting of different annotators.

The final labeled text is as follows:

Launching O
the O
sixth O
branch O
of O
the O
US B-ORG
armed I-ORG
forces I-ORG
Defense I-ORG
News O

4. EXPERIMENT

In this section, we compare our method with existing baseline methods and demonstrate the effectiveness of our method by experimental results.

4.1. Experimental Setup

On the NER basic dataset, we use CoNLL-2003 dataset, which mainly contains four types of labels, namely PER (persons), LOC (locations), ORG (organizations), MISC (miscellaneous entities). The text in science and technology equipment domain is used as the dataset in the field of resource scarcity, and the characteristics of the text in this field are comprehensively analyzed. As mentioned in the preceding section, we mainly annotate six entities, namely PER (persons), ORG (organizations), PCT (products), OUT (Outcomes), SER (Services), TIM (time), etc. The experimental evaluation index is the F1 score of named entity recognition.

In terms of network structure setting, the standard of LM-LSTM-CRF model is mainly followed. The hyper-parameters of character-level LSTM and word-level LSTM are the same. The size of

hidden layer is set to 300, the dimension of character-level embedding layer is 30, the dimension of word-level embedding layer is 100, and the depth of highway layer is set to 1.

As for the training optimization, the small-batch stochastic gradient descent method combined with momentum is used. The batch size and momentum are set to 10 and 0.9, respectively. The learning rate is set to $\eta_t = \frac{\eta_0}{1 + \rho t}$, where η_0 is the initial learning rate and $\rho=0.05$ is the decay ratio.

In order to alleviate the over-fitting problem, the proportion of Dropout is fixed to 0.5. In order to resolve the gradient explosion problem that may occur in the training process, the gradient clipping strategy is adopted, and the threshold of gradient clipping is set to 5.0, when the gradient is updated, if the gradient vector is greater than this threshold, it will be limited in this range.

4.2. Results and Analysis

Our experiments are mainly compared with two baselines, the LM-BiLSTM-CRF model as the experimental basis, “LBC” for short, and the BERT-LSTM-CRF model with good performance in the named entity recognition task in recent years, “BBC” for short.

(1) Adding domain dictionary

First, we run the experiment of adding a domain dictionary, we call the LM-BiLSTM-CRF model with dictionary as Dicstrengthen-LM-BiLSTM-CRF model, “DLBC” for short. The training data used here comes from the Conll2003 data set. When training the DLBC model, use the dictionary of science and technology equipment mentioned before, and integrate its information into the word vector representation to obtain the target model. We use LBC model, BBC model and DLBC model for our experiments on the text in the field of scientific and technological equipment, and the results are shown in Table 1.

Table 1. F1-score (%) of three model

Model	PER	LOC	ORG	OVERALL
LBC	80.32	81.25	79.57	80.38
BBC	83.88	82.54	83.06	83.16
DLBC	85.72	81.53	85.21	84.15

It can be seen from the results in the table that the F1 value of each model is generally not high, which is mainly because the training data used by these models are from the Conll2003 data set, and the field characteristics mastered in the training are not suitable for the field of scientific and technological equipment. Comparing the results of each model, it can be seen that the DLBC model with dictionary has achieved the best results in the name entity PER, the organization entity ORG and overall F1, especially compared with the LBC model without dictionary. However, in terms of location entity recognition, the DLBC model is less effective than the BBC model, basically similar to the LBC model, mainly because of the low priority of location entities in the field of science and technology equipment, and the dictionary we collect does not contain many location entities, so the results are basically the same as when no domain dictionary is added. Overall, adding domain dictionaries can effectively improve the recognition effect of named entity recognition model.

(2) Adding small batch labeled data

Next, we test the effect of adding small batch labeled data on the experimental results. The LM-BiLSTM-CRF model using small batch labeled data is called LM-BiLSTM-CRF-annplus,

“LBC+” for short. The small batch labeled data used are provided by the concentrated annotation of some experts in a short time. The original texts are news texts from the field of science and technology equipment. This experiment will explore the relationship between the size of labeled text and model results, as shown in Figure 1 and Figure 2.

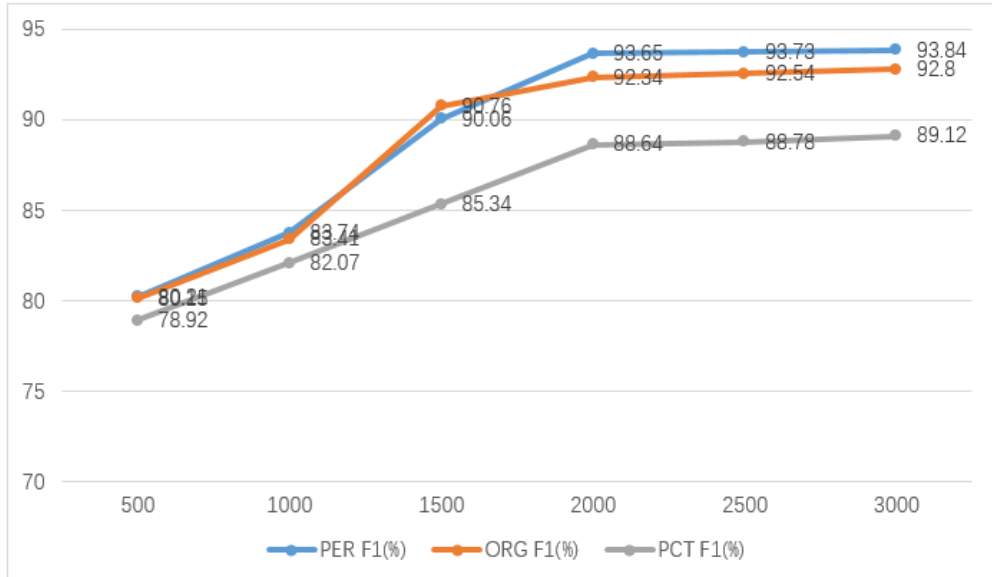


Figure 1. Named Entity Recognition Results Varying with Data Scale 1

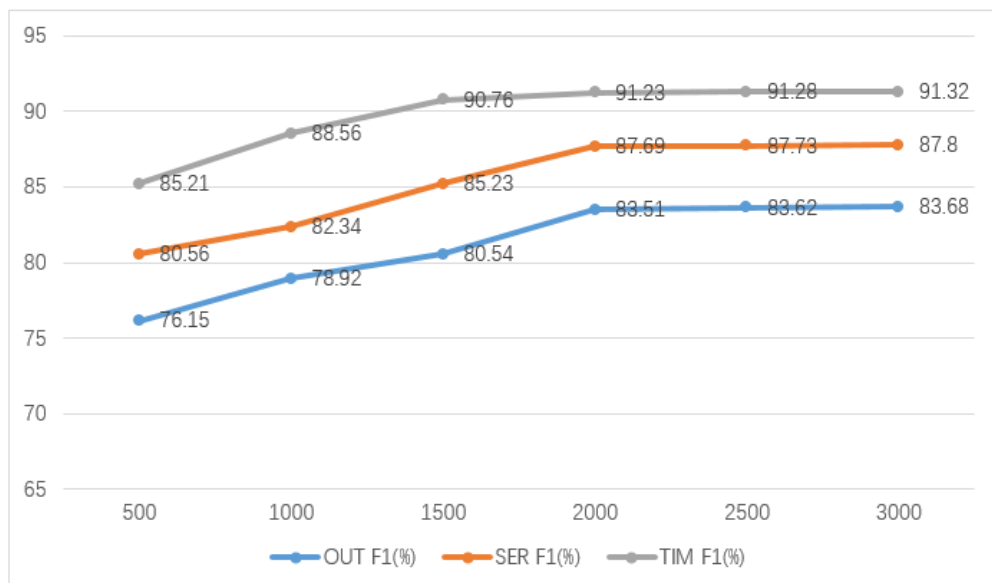


Figure 2. Named Entity Recognition Results Varying with Data Scale 2

These two figures show the change of entity recognition results with the scale of labeled data. Here we define six entities, namely PER (name), ORG (organization), PCT (product), OUT (result), SER (position) and TIM (time). It can be seen from the figure that at the beginning, when the scale of labeled documents gradually increases, the results of entity recognition also increase. However, when the scale of labeled documents reaches more than 2000, even if the

labeled documents continue to increase, the F1 value basically no longer increase or only slightly increase. Considering the increase of annotation cost, we finally decide that the appropriate scale of labeled documents should be set at 2000.

In addition, it can be seen from Figure 2 that compared with other types of entities, the recognition results of TIM (time) do not improve significantly with the increase of the scale of labeled data, which is mainly because the form of time entities in the text is relatively fixed. Even if we increase the scale of labeled documents, the model cannot obtain more time entity features.

(3) Adding domain dictionary and small batch labeled data

Finally, we add the domain dictionary and small batch annotation data to the model at the same time, and obtain our final model Dicstrength-LM-BiLSTM-CRF-annplus, “DLBC+” for short. The experimental results of this model compared with other baseline models are shown in Table 2.

Table 2. F1-score (%) of three model

Model	PER	ORG	PCT	OUT	SER	TIM	OVERALL
LBC+	93.65	92.34	88.64	83.51	87.69	91.23	89.51
BBC+	94.15	92.89	90.71	85.29	89.36	95.08	91.25
DLBC+	96.24	94.37	91.75	87.65	88.14	96.76	93.05

Based on the above experimental results, it can be seen that the LM-LSTM-CRF model combined with the dictionary is better than LM-LSTM-CRF model and BERT-LSTM-CRF model on the whole, because the domain-specific name information contained in the dictionary improves the recognition rate in general, word embedding with dictionary enhancement carries more useful information for recognition. Then, the model using domain labeled data as training data performs significantly better than the model using CoNLL-2003 dataset as training data in terms of person and organization, which is also well understood, because the domain information contained in the domain data is much more than that contained in the general data set, through the study of the common information in the field, the model grasps more domain-specific implicit features and so as to obtains better results.

In addition, compared with the improvement of ‘DLBC+’ on ‘LBC+’, the improvement of ‘DLBC’ on ‘LBC’ is more obvious. This is because when the domain dictionary is applied to the model with domain labeled data as the training set, some domain proper names have been labeled, that is, some information in the dictionary has been used, so the improvement effect is relatively less obvious. Finally, ‘BBC+’ is superior to ‘DLBC+’ in the recognition of SER (service), which may be due to the fact that some ambiguous words in the dictionary disturb the experimental results, and also represent that the quality of the current dictionary can be improved.

Although the experimental results show that our model can solve the problem of named entity recognition in many low resource domains, there are still some limitations and deficiencies. It depends on the quality of the domain dictionary, if there is a lack of high-quality domain dictionaries in the field to be processed, then we will need to invest more costs in domain data annotation to achieve better processing results, thereby increasing manpower consumption. We plan to improve this problem in our further research in the future.

5. CONCLUSIONS

We propose a method to improve named entity recognition in low resource domain using domain knowledge, and present its effectiveness in the field of science and technology equipment. The experimental results show that our method has achieved competitive results in low resource domain, and F1 score has been significantly improved compared with other traditional baseline methods.

In addition, on the basis of the existing work, our next steps mainly include:

- (1) Experiments in more resource scarce fields, and developing corresponding processing schemes for domain characteristics to improve the model processing effect.
- (2) Apply the dictionary-based approach to more existing models to see if we can improve their performance.
- (3) Conduct experiments on larger datasets to observe the improvement of model performance and explore the most appropriate training dataset size for our model.

ACKNOWLEDGEMENTS

This research work is financially supported by The National Key Research and Development Program of China (NO.2017YFB1002101). This work was also funded by the Institute of Science and Development, Chinese Academy of Sciences (NO.GHJ-ZLZX-2020-42) and the Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (No. 6141B08010102). The authors want to thank for the helpful suggestions from Heyan Huang, Chong Feng, Bo Wang and Ximo Bian. The corresponding author of this article is Yuan Shi.

REFERENCES

- [1] Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354.
- [2] Akbik, A., Blythe, D., & Vollgraf, R. (2018, August). Contextual string embeddings for sequence labeling. In Proceedings of the 27th international conference on computational linguistics (pp. 1638-1649).
- [3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- [5] Strubell, E., Verga, P., Belanger, D., & McCallum, A. (2017). Fast and accurate entity recognition with iterated dilated convolutions. arXiv preprint arXiv:1702.02098.
- [6] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- [7] Li, Q., Li, H., Ji, H., Wang, W., Zheng, J., & Huang, F. (2012, October). Joint bilingual name tagging for parallel corpora. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 1727-1731).
- [8] Yang, Z., Salakhutdinov, R., & Cohen, W. W. (2017). Transfer learning for sequence tagging with hierarchical recurrent networks. arXiv preprint arXiv:1703.06345.
- [9] Chen, P., Ding, H., Araki, J., & Huang, R. (2021, January). Explicitly Capturing Relations between Entity Mentions via Graph Neural Networks for Domain-specific Named Entity Recognition. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers).
- [10] Liu, L., Shang, J., Ren, X., Xu, F., Gui, H., Peng, J., & Han, J. (2018, April). Empower sequence labeling with task-aware neural language model. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

- [11] Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Highway networks. arXiv preprint arXiv:1505.00387.
- [12] Ma, R., Peng, M., Zhang, Q., & Huang, X. (2019). Simplify the usage of lexicon in Chinese NER. arXiv preprint arXiv:1908.05969.
- [13] Zhang, Y., & Yang, J. (2018). Chinese NER using lattice LSTM. arXiv preprint arXiv:1805.02023.
- [14] Zhao, H., & Kit, C. (2008). Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing.
- [15] Peng, N., & Dredze, M. (2016). Improving named entity recognition for chinese social media with word segmentation representation learning. arXiv preprint arXiv:1603.00786.
- [16] S Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.

Author

Shi Yuan, born in 1995, master of science in cyberspace security at School of Computer Science & Technology Beijing Institute of Technology. His main research interest is named entity recognition. (18810932070@163.com)



AN OBJECT-DRIVEN COLLISION DETECTION WITH 2D CAMERAS USING ARTIFICIAL INTELLIGENCE AND COMPUTER VISION

Yang Liu¹, Evan Gunnell², Yu Sun², Hao Zheng³

¹Department of Mechanical and Aerospace Engineering,
George Washington University, Washington, DC 20052

²California State Polytechnic University, Pomona, CA, 91768

³ASML, Wilton, CT 06897

ABSTRACT

Autonomous driving is one of the most popular technologies in artificial intelligence. Collision detection is an important issue in automatic driving, which is related to the safety of automatic driving. Many collision detection methods have been proposed, but they all have certain limitations and cannot meet the requirements for automatic driving. Camera is one of the most popular methods to detect objects. The obstacle detection of the current camera is mostly completed by two or more cameras (binocular technology) or used in conjunction with other sensors (such as a depth camera) to achieve the purpose of distance detection. In this paper, we propose an algorithm to detect obstacle distances from photos or videos of a single camera.

KEYWORDS

Autonomous driving, computer vision, machine learning, artificial intelligence, distance detection, collision detection.

1. INTRODUCTION

As a new product of modern society, artificial intelligence [1] is the future direction of development. It can operate automatically in a specific environment according to a present mode. Without human management, the expected or higher goals can be achieved. Autonomous driving[2] is one of the most popular technologies in artificial intelligence. It can bring great convenience to our lives, and at the same time it is a kind of release to people's fatigue when driving. From the perspective of the nature of autonomous driving, it is essentially a fast-reacting robot, and its level of intelligence is relatively high. If autonomous driving can be achieved, it will mean a huge step forward not only in commuting but also in the field of robotics, as the same level of artificial intelligence can be applied to more standard robots. What's more, autonomous driving technology can provide people with a huge benefit of time. After being released from the requirement to drive, people in the car can do what they want. At this time, the car on the journey is a brand-new space for people's lives, and people will have one more means of living their life than before. The supporting facilities of the car will be completely changed, and passengers can work and entertain. This change in the sense of space provides a new service model and a new life experience for social development. Collision detection [3] is an important issue in automatic driving, which is related to the safety of automatic driving.

Nowadays, many collision detection methods have been proposed, but they all have certain limitations and cannot meet the requirements for automatic driving. Ultrasound [4] is an important means of collision detection. The energy consumption of ultrasonic waves is relatively slow, the propagation distance in the medium is relatively long, the realization is convenient, the cost is low, the calculation is simple, and it is easy to achieve real-time control. Ultrasonic radar has great advantages in short-distance measurement. However, ultrasonic radar has certain limitations in measuring distance at high speeds and is greatly affected by the weather. Moreover, the propagation speed of ultrasonic waves is slow, and when the car is running, it cannot keep up with the change of the distance between the cars in real time. In addition, the ultrasonic scattering angle is large, and the directivity is poor. When measuring a long-distance target, its echo signal will be relatively weak, which affects the measurement accuracy.

Another common method is the camera [5]. The camera is generally composed of a lens, an image sensor, an Image Signal Processor (ISP) [6], and a serializer. The general procedure is that the basic information of the object collected by the lens is processed by the image sensor and then sent to the ISP for serialized transmission. Transmission methods can also be divided into LVDS-based transmission on coaxial cable or twisted pair or direct transmission via Ethernet. The camera is mainly used in the automatic driving system for obstacle detection, lane line detection, road information reading, map construction and auxiliary positioning. However, the obstacle detection of the current camera is mostly completed by two or more cameras (binocular technology) [7] or used in conjunction with other sensors (such as a depth camera [8]) to achieve the purpose of distance detection.

The depth camera based on binocular stereo vision is similar to the human eyes, and is different from the depth camera based on TOF [9] and structured light principle. It does not actively project the light source to the outside, and completely relies on the two pictures taken (color RGB or grayscale) to calculate the depth, so it is sometimes called a passive binocular depth camera.

TOF is short for Time of flight, literally translated as the meaning of flight time. The so-called time-of-flight method 3D imaging is to continuously send light pulses to the target, and then use the sensor to receive the light returning from the object, and obtain the target object distance by detecting the flight (round trip) time of the light pulse. This technology is basically similar to the principle of the 3D laser sensor [10], except that the 3D laser sensor scans point by point, while the TOF camera obtains the depth information of the entire image at the same time.

All of these technologies are often combined into one unit with the data being passed through some level of advanced object detection [11] or other machine learning algorithm. However, our aim in this paper is to reduce the number of elements needed to produce an accurate prediction so as to reduce the barrier to entry when utilizing automated object detection.

Generally, humans obtain depth information through their eyes. Binocular technology in particular is based on this idea. However, even when we somewhat limit our view and observe something with only one eye, we can still feel the distance of the object. Even with some loss of depth perception, humans can generally still accurately predict distances with one eye closed. This is because people themselves have a very good understanding of the world where they live (prior knowledge). They have a basic prediction of the size of everyday objects (visual training for many years) [12]. According to common sense, it is indeed possible to infer the distance of the object in the image. In addition, when a person observes an object with a single eye, the human eye is actually frequently moving and scanning its surroundings. This is functionally equivalent to a moving monocular camera [13], which is similar to the principle of the structure from motion. The moving monocular camera compares the difference of multiple frames. It is indeed possible to get in-depth information. It shows that humans can also obtain a certain depth

of information from a single eye. At the same time, the depth information is obtained by comparing the differences of multiple frames. Therefore, it stands to reason that it is feasible to obtain depth information with just a single camera. In this paper, we propose an algorithm to detect obstacle distances from photos or videos of a single camera.

In this project, we did three experiments to finish the distance detection function. First of all, we added distance labels in YOLOv5 [14] so that we can get the real-time information from the screen. Then we collected the dataset of one object to build and test our models. After we prove that the distance information can be obtained. We collected the dataset of different objects to compare the results of different models. Lastly, different cameras were selected to collect the dataset, in order to explore and validate the impact of different camera types.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, followed by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

Challenge 1: Learning and training the model to predict the distance for specific objects shown in the view based on the size (dimension) of the recognized object is difficult. In different environments, various factors such as the type, appearance, and size of the recognition object are different. For example, in the case of autonomous driving, even in different countries and regions, there are still differences in objects with uniform standards such as traffic lights, buses, and taxis. Therefore, the first step of training and learning, establishing a database, is a big challenge. Although there are already some object recognition databases, for pedestrians, animals, and other objects that have individual differences, the addition of size information data is still a big challenge. For example, the height of an adult man may range from 1.65m to 1.90m, so more detailed classification is needed. In addition, the camera parameters and camera postures in each model are different. Whether this factor will affect the distance detection needs to be further explored. Although in theory a monocular camera can obtain a certain information for depth, it is still doubtful whether the information that is obtained by a monocular camera can meet the requirements for automatic driving.

Challenge 2: The impact on the accuracy of the distance prediction coming from the different camera types is unknown. There are many types of cameras on the market today. The influence of various camera parameters on object recognition and distance detection is unclear. It is also a big challenge to find the influence of the parameters of each camera on the distance detection. Among the various parameters, the change of the focal length will inevitably affect the distance detection, so during use, how much influence the change of the focal length will have on the distance detection remains to be explored. This will determine whether to use a zoom camera or a fixed focus camera in the end. Aperture is another important parameter of the camera, which directly contributes to the sharpness of the image. The aperture must have a certain influence on object recognition, but whether it has a greater influence on distance detection still needs to be explored. So how to choose the type of camera is also a big challenge. Due to how many camera options there are, we will simply have to select one type of camera and use that as the standard for all of our testing.

Challenge 3: Choosing a reference machine learning model [12] involves multiple issues. Relying on the existing reference model to predict the distance of other objects in the same view

is a possible method to solve our problem. There are many related models for object recognition. On this basis, adding size data information to establish a mapping relationship with depth is the main task of this paper. In the object recognition model, the reliability of object recognition is also a major factor affecting distance detection. So how to choose a reference model is also a big challenge. Measuring data input values and model selection accuracy can help us pick the most optimal reference model. However, it would take a lot of time to test every model. We can read some related work about these models so that we know the range of their application. We can also learn their advantages and disadvantages from this work. From the studies that already exist in these papers, we can choose some of the models which might be better than others. Then we can use the same input data, which we know all the information, to test each model we chose and choose the one that gives us the highest accuracy. To measure the accuracy of our model we will provide a dataset of known objects and distances and see how well it performs. We can then also tune the model once it is chosen to tailor it specifically to our problem.

3. METHODOLOGY

In this section, we will introduce the Overview of the System and Models and Feature Selections.

3.1. Overview of the System

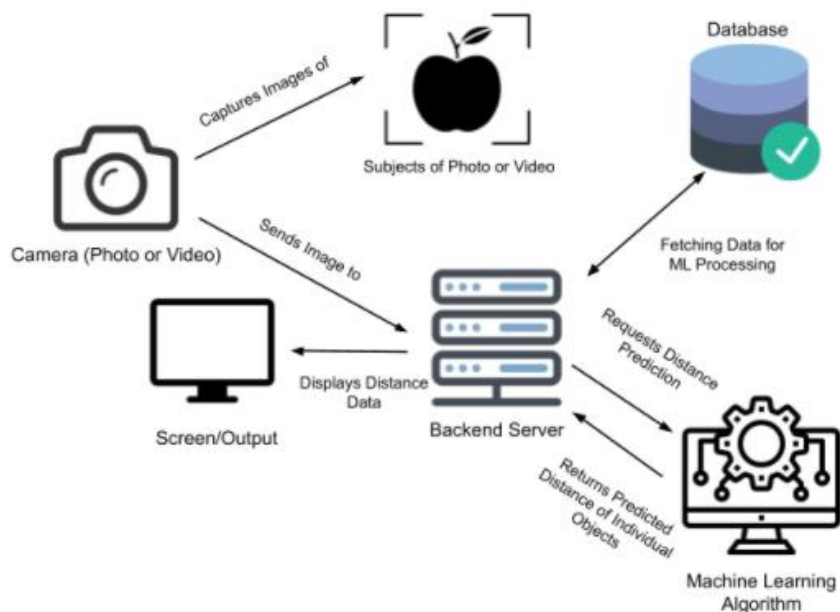


Figure 1. The Overview of the System Architecture

The proposed system framework is shown in the Figure 1. The input can be submitted in two ways: one is to take photos captured by the camera as input, and the other being to use video as input. After getting the specified data, the program will use a Web API/Web service to handle communication between the frontend, the database and the backend server. This can be implemented in a service such as Python Flask. The database we are using in particular is a non-relational database that holds all of our training data and testing data. This data is utilized to obtain the distance information through the machine learning algorithm and heuristic. Aside from just training data, it also contains important information related to the approximate size of certain objects to help better guide the system. The machine learning portion of the system uses modern regressive machine learning libraries. We will show some testing results for a variety of the

common models such as Linear Regression, Polynomial regression, and Random Forest Regression.. After undergoing all of this processing in our backend web server, the picture or video with the distance information is returned to the display screen as output. If it is a video, it dynamically updates the screen with the most recent distance information. For a picture, it will simply estimate the current distance for the recognized objects.

3.2. Models and Feature Selections

The primary focus of the project is the machine learning component, with us a common YOLO machine learning model. YOLO, standing for “You Look Only Once”, is a member of the object detection model family. Its first iteration was released in 2016 by Joseph Redmon with multiple modern iterations being implemented. This particular project focuses on the YOLOv5 model, augmenting it with additional machine learning to accomplish our project task of estimating distance of recognized objects. The YOLO family of models consists of three main architectural blocks: the Backbone, the Neck, and the Head. Each component of the models helps to employ a specific feature of it. Combined they allow the model to quickly compute object detection on a variety of objects. The backbone employs CSPDarknet as the primary tool for image feature extraction consisting of cross-stage partial networks. The neck represents the aggregation of training features that will later be used by the head. It uses PANet to generate a feature pyramid network for its aggregation. Lastly, the YOLOv5 head focuses on providing multiple layers that generate predictions from the anchor boxes for object detection.[14]

Our project aims to extend its fundamental object recognition moduel by adding distance information in both photo and/or video predictions. Below is an example of some of the code added to the library.

```
def calculate_distance(xyxy):
    img_height = 640
    img_width = 640

    # Calculate the relative size of the the bounding box for a person. 80% = 2ft, 50% = 5ft
    percentage_of_screen = ((xyxy[2]-xyxy[0]) / img_width) * 100

    # 1ft = 90% +
    # 2ft = 80% screen (0.78)
    # 3ft = 70%
    # 4ft = 60%
    # 5ft = 50% screen (0.5)
    # 6ft = 40%
    # 7ft = 30%
    # 8ft = 22.5% screen (0.225)
    # print(percentage_of_screen)
    print((xyxy[2]-xyxy[0]) / img_width)
    print("LOOKING AT A PERSON")

    return "L = " + str((100 - percentage_of_screen) / 10)
```

Figure 2. The Code Excerpt of Calculating the Distance using a Base Formula

In this paper, we present an algorithm based on the width of the object to get the distance information. We can judge the distance by the width of screen ratio of the object, because when the object goes farther away from the camera, the width of screen ratio of the object will be smaller. For the similar reason, we could also use the height of screen ratio to get the depth information when the width of the object is too small to be measured or exceeds the range of the camera. Utilizing width or height fixes an issue of potential camera angles. When a camera is

higher or lower affects the estimated height while if it is to the left or right of the object it distorts the perceived width. Once we collect the data of the width of image ratio and the distance between object and camera, we could build a model reflecting the relationship between the percentage of the screen and the distance of the current object. To build the model, we could use machine learning algorithms [12] in the width-based detection. Results of testing the various predictive models will be available in Section 4 later in the paper.

Because we get the distance information from the width of screen ratio, camera wide angle will be the most direct factor to influence the distance prediction. Changing the total width view of the camera has a drastic impact on what the image will look like, in turn distorting the potential results of the object detection. For example, a fish-eye camera lens has a significantly larger viewing angle than a standard phone camera does. The focal length also will be a factor to influence the distance prediction. As the focal length decreases, the width view will be larger, and the distortion of the picture will also become larger. Two same things with different distances in photos, the closer one will be much larger than the farther one. However, the closer one in shorter focal length cameras will seem larger than the one in longer focal length cameras. Another factor for cameras which will influence the distance prediction is the aspect ratio of the photo or video. The different aspect ratio will distort the object in video or photo. The shape of the object will be changed. For example, when we watch TV, if the aspect ratio is 4:3, the people will appear shorter and fatter than the aspect ratio is 16:9.

In order to better simulate the effect of human monocular distance measurement, we think that it is better to use video than photos. Due to the existence of waves of movement, the comparison between each frame and the previous frame or several frames can theoretically achieve the effect of binocular imaging. As a result, a 3D space can be better constructed, so that the distance information of the object can be obtained better and more accurately.

After selecting the appropriate model, we can use the information from our database to train the model and get it prepped for potential input. It is then integrated with the rest of the application. It will connect to the API of the backend server, take in potential image/video data, and then return the most accurate distance estimate to the photo or video.

4. EXPERIMENTS

Three experiments have been designed and conducted to illustrate the performance of the proposed distance prediction algorithm.

4.1. Experiment 1: Distance Estimation using Machine Learning Models

In this part, we choose several machine learning models to finish distance prediction. We collect the width percentage of the screen with different distances from a person to the camera.

By using this data to predict the distance, we choose the Linear Regression model [15], Polynomial Ridge Regression model [16], Random Regression model [17], and ElasticNet Regression model [18]. For the Polynomial Ridge Regression model, we select different Poly Features, such as 3 Poly Features and 4 Poly Features. For the Random Regression model, we compare 2 max depth models and no max depth models.

4.2. Experiment 2: The Impact of the Object Type on the Distance Estimation

In this part, we choose several machine learning models to finish distance prediction and compare the influence of object type. Besides the data we collect in experiment 1, we also collect the data of different objects, such as the cell phone and stuffed penguin.

By using this data to predict the distance, we choose the Linear Regression model [15], Polynomial Ridge Regression model [16], Random Regression model [17], and ElasticNet Regression model [18]. For the Polynomial Ridge Regression model, we select different Poly Features, such as 3 Poly Features and 4 Poly Features. For the Random Regression model, we compare 2 max depth models and no max depth models.

4.3. Experiment 3: The Impact of the Camera Type on the Distance Estimation

In this part, we choose several machine learning models to finish distance prediction and compare the influence of camera type. Besides the data we collect in experiment 1 and 2, we also collect the data of the second person using a different camera (Dell G5).

By using this data to predict the distance, we choose the Linear Regression model [15], Polynomial Ridge Regression model [16], Random Regression model [17], and ElasticNet Regression model [18]. For the Polynomial Ridge Regression model, we select different Poly Features, such as 3 Poly Features and 4 Poly Features. For the Random Regression model, we compare 2 max depth models and no max depth models.

4.4. Dataset and Results

As Figure 3 shows the camera information, the object width information and the input data, Figure 4 shows the distance data as the output data, test input data and the machine learning model we use. Figure 5 shows the output of the test input data, including predicting distance and the Cross Validation Average Testing Scores.

```

# 1. change the model (Linear Regression, Polynomial Regression, RandomForestRegression)
# 2. same model, change the parameters

# Data [(camera type), object, width]

#Evan Camera: Razer Kiyo ---camera 0
#Still image: 4 megapixel
# Video: 1080p at 30 fps, 720p at 60 fps

#Object width: 0 - person (20in.), 1 - cell phone (6in. horizontal), 2 - stuffed penguin (13in.)
# Yang Camera: Dell G5 ---camera 1
# Still image: 0.92 megapixel (HD)
# Video: 1280 x 720 (HD) at 30 fps
#Object width: 0 - person (15.5in.), 1 - cell phone (6in. horizontal)|
input_data = [
[0, 0, 80],
[0, 0, 60],
[0, 0, 46.5],
[0, 0, 40],
[0, 0, 35],

[0, 1, 56],
[0, 1, 28],
[0, 1, 18],
[0, 1, 14],
[0, 1, 11.75],
[0, 1, 8.5],

[0, 2, 65],
[0, 2, 55],
[0, 2, 47],
[0, 2, 41.2],
[0, 2, 30],
[0, 2, 20],
[0, 2, 15],

[1, 0, 67],
[1, 0, 46],
[1, 0, 40],
[1, 0, 32.5],
[1, 0, 28],

[1, 1, 54],
[1, 1, 26],
[1, 1, 16.7],
[1, 1, 13],
[1, 1, 10.5],
[1, 1, 7]

```

Figure 3. The Input Dataset for the Experiments

```

#Distance in feet
output_data = [
    2,3,4,5,6, # person 1
    1,2,3,4,5,6, # phone 2
    2,3,4,5,6, 7, 8, # stuffed penguin
    2,3,4,5,6, # person 2
    1,2,3,4,5,6, # phone 2
]

input_data.extend(input_data)
input_data.extend(input_data)

output_data.extend(output_data)
output_data.extend(output_data)

input_data.extend(input_data)
input_data.extend(input_data)

output_data.extend(output_data)
output_data.extend(output_data)

test = [[0, 1, 20]]
cv_num = 5

# Linear Regression
model = linear_model.LinearRegression()
model.fit(input_data, output_data)
print(model.predict(test))
print(sum(cross_val_score(model, input_data, output_data, cv = cv_num, scoring = 'neg_mean_squared_error')) / cv_num)

# Polynomial Ridge Regression (4 Poly Features)
model2 = make_pipeline(PolynomialFeatures(4), linear_model.LinearRegression())
model2.fit(input_data, output_data)
print(model2.predict(test))
print((cross_val_score(model2, input_data, output_data, cv = cv_num)))
print(sum(cross_val_score(model2, input_data, output_data, cv = cv_num) / cv_num))

# Polynomial Ridge Regression (3 Poly Features)
model2_5 = make_pipeline(PolynomialFeatures(3), linear_model.LinearRegression())
model2_5.fit(input_data, output_data)
print(model2_5.predict(test))
print(sum(cross_val_score(model2_5, input_data, output_data, cv = cv_num, scoring = 'neg_mean_squared_error')) / cv_num)

# RF Max_depth = 2
model3 = RandomForestRegressor(max_depth = 2, random_state = 0)
model3.fit(input_data, output_data)
print(model3.predict(test))
print(sum(cross_val_score(model3, input_data, output_data, cv = cv_num, scoring = 'neg_mean_squared_error')) / cv_num)

# RF No max Depth
model3_5 = RandomForestRegressor(random_state = 0)
model3_5.fit(input_data, output_data)
print(model3_5.predict(test))
print(sum(cross_val_score(model3_5, input_data, output_data, cv = cv_num, scoring = 'neg_mean_squared_error')) / cv_num)

# ElasticNet Regression
model4 = ElasticNet(random_state = 0)
model4.fit(input_data, output_data)
print(model4.predict(test))
print(sum(cross_val_score(model4, input_data, output_data, cv = cv_num, scoring = 'neg_mean_squared_error')) / cv_num)

```

Figure 4. The Code Excerpt for Dataset Training and Cross Validation

```

[5.27190273]
-1.8569599858280637
[2.71911058]
[0.99693576 0.99663323 0.99688429 0.99680586 0.99652266]
0.9967563586641514
[2.89022235]
-0.02830028786354614
[4.29479788]
-1.4191690835273607
[3.]
0.0
[4.83249015]
-2.034562505455932
[0.47311828 0.43010753 0.43010753 0.44086022 0.43478261]
0.4417952314165498
[0.03019402 0.17018951 0.79961646]
[0.5483871 0.53763441 0.55913978 0.62365591 0.51086957]
0.5559373539036933
[0.01038713 0.05466782 0.93494505]
[1. 1. 1. 1. 1.]
1.0
.

```

Figure 5. The Excerpt of the Test Results

Figure 6 shows the Cross Validation Average Testing Scores for different model. From Figure 6, we can know that Random Forest (depth=2) get the highest score and ElasticNet Regression get the lowest score. In this way, we could say Random Forest (depth=2) is the most suitable model for the data.

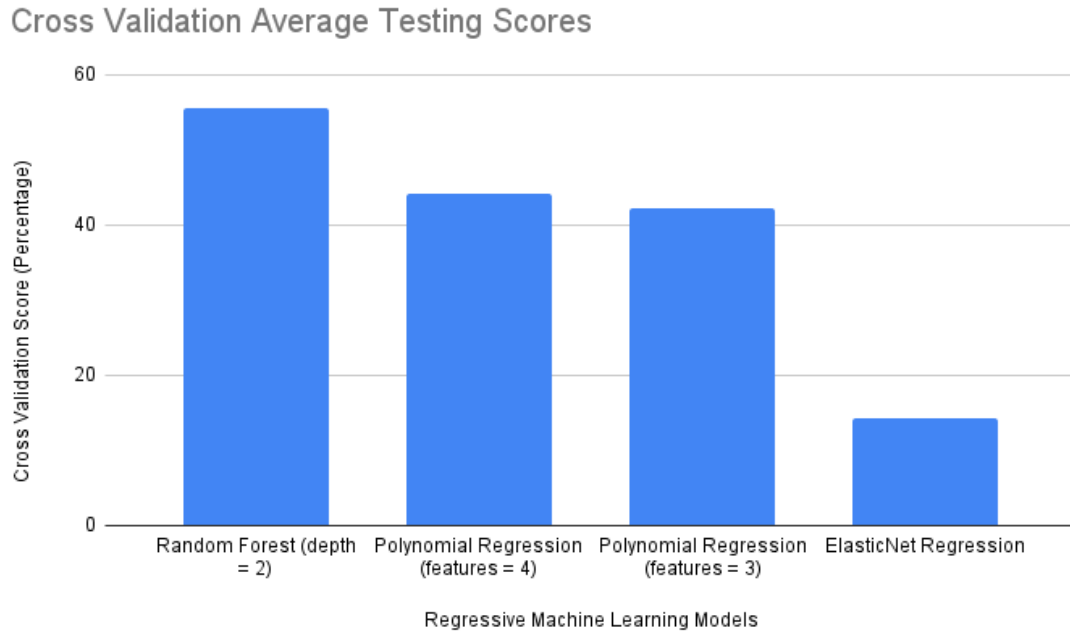


Figure 6. The Accuracy Comparison of Different Machine Learning Models

Figure 7 shows the influence of various factors on distance prediction. We can know that the percentage of screen width is the most important factor to predict the distance. The camera and the object has less influence on prediction.

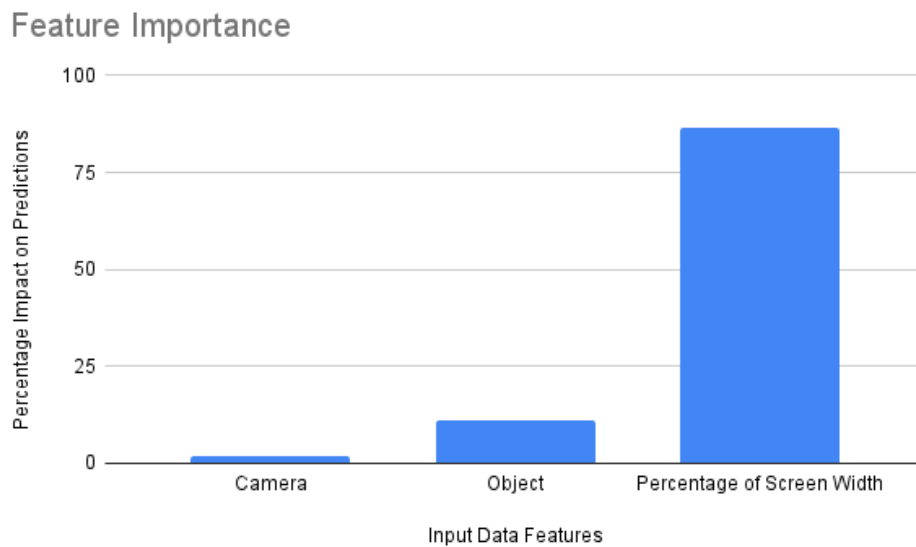


Figure 7. The Impact Comparison with the Different Extra Feature Set

5. RELATED WORK

Iro Laina. et al proposed a fully convolutional architecture to model the ambiguous mapping between monocular images and depth maps. They use MatConvNet, and train on a single NVIDIA GeForce GTX TITAN with 12GB of GPU memory. The network is trained on RGB inputs to predict the corresponding depth maps. They use data augmentation to increase the number of training samples. They model small translations by random crops of the augmented images down to the chosen input size of the network [19]. Compared to our work, they don't get the actual distance but a relative location, while we will point out the exact distance from the camera to the objects.

Michael W. Tao. et al presented a defocus and correspondence algorithm to get the depth information. Their algorithm comprises three stages. The first stage is to shear the EPI and compute both defocus and correspondence depth cue responses. The second stage is to find the optimal depth and confidence of the responses. The third stage is to combine both cues in a MRF global optimization process [20]. However, compared to our work, their algorithm relies on shearing, so that objects that are too far from the main lens's focal plane will have incorrect depth estimations. They also rely on Lytro consumer cameras but we are able to apply in more general cameras.

Mansour, M. et al compared depth estimation performance of motion parallax and binocular disparity visual cues by using two different camera resolutions and feature points locations. Mansour, M. et al also proposed a method to overcome the limitations of the stereo camera by switching between motion parallax and binocular disparity [21]. Compared to our work, their work depends on motion parallax, and binocular disparity visual, which depends on at least two cameras. We are able to get depth information from still photos or a video from a single camera.

6. CONCLUSIONS AND FUTURE WORK

In this project, we proposed an algorithm which is designed to detect the distance of obstacles from a photograph or video from a single camera. The model identified distances for specific objects based on the size (dimension) of an object. We used different machine learning models to predict the distance, and compare the results of different models. We found that the object which is closer from the camera, the width percentage will change quickly. While it is farther, the change will become slower and even there is no change after a certain distance. It is obvious for small size objects. The smaller the object is, the earlier there is no change. For model choosing, we found the Polynomial Ridge Regression model (neither 3 Poly feature or 4 Poly feature) does not work well. The Linear Regression model works better than the Polynomial Ridge Regression model. It predicts correctly for close objects but for farther, it has more room to improve. The Random Regression and ElasticNet Regression model works best in these models.

In addition, one limitation in this project is that it does not suggest the sufficient threshold of training dataset. One thing we plan to improve is to evaluate the accuracy of the training process and collect more dataset to improve the accuracy.

As for the future work, we will investigate other machine learning algorithms to keep improving accuracy of the distance prediction. We also would like to explore the possibility of applying deep learning [22] in this problem domain. We will also continue to study the impact of camera parameters on predictions and select the most suitable camera to use.

REFERENCES

- [1] Jackson, Philip C. Introduction to artificial intelligence. Courier Dover Publications, 2019.
- [2] Levinson, Jesse, et al. "Towards fully autonomous driving: Systems and algorithms." 2011 IEEE intelligent vehicles symposium (IV). IEEE, 2011.
- [3] Hubbard, Philip Martyn. "Collision detection for interactive graphics applications." IEEE Transactions on Visualization and Computer Graphics 1.3 (1995): 218-230.
- [4] Mason, Timothy J., et al. "Application of ultrasound." Emerging technologies for food processing. Academic Press, 2005. 323-351.
- [5] Sturm, Peter, and Srikumar Ramalingam. Camera models and fundamental concepts used in geometric computer vision. Now Publishers Inc, 2011.
- [6] Wu, Chyuan-Tyng, et al. "VisionISP: Repurposing the image signal processor for computer vision applications." 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019.
- [7] Cao, Zhi-Le, Zhong-Hong Yan, and Hong Wang. "Summary of binocular stereo vision matching technology." Journal of Chongqing University of Technology (Natural Science) 29.2 (2015): 70-75.
- [8] Izadi, Shahram, et al. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." Proceedings of the 24th annual ACM symposium on User interface software and technology. 2011.
- [9] Remondino, Fabio, and David Stoppa, eds. TOF range-imaging cameras. Vol. 68121. Heidelberg, Germany: Springer, 2013.
- [10] Konolige, Kurt, et al. "A low-cost laser distance sensor." 2008 IEEE international conference on robotics and automation. IEEE, 2008.
- [11] Viola, Paul, and Michael Jones. "Robust real-time object detection." International journal of computer vision 4.34-47 (2001): 4.
- [12] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. "Machine learning basics." Deep learning 1.7 (2016): 98-164.
- [13] Haseeb, Muhammad Abdul, et al. "DisNet: a novel method for distance estimation from monocular camera." 10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS (2018).
- [14] Introduction to YOLOv5 Object Detection with Tutorial, <https://machinelearningknowledge.ai/introduction-to-yolov5-object-detection-with-tutorial/>, 2021
- [15] Yao, Weixin, and Longhai Li. "A new regression model: modal linear regression." Scandinavian Journal of Statistics 41.3 (2014): 656-671.
- [16] Cheng, Xi, et al. "Polynomial regression as an alternative to neural nets." arXiv preprint arXiv:1806.06850 (2018).
- [17] Misztal, Ignacy. "Properties of random regression models using linear splines." Journal of Animal Breeding and Genetics 123.2 (2006): 74-80.
- [18] Hans, Chris. "Elastic net regression modeling with the orthant normal prior." Journal of the American Statistical Association 106.496 (2011): 1383-1393.
- [19] Laina, Iro, et al. "Deeper depth prediction with fully convolutional residual networks." 2016 Fourth international conference on 3D vision (3DV). IEEE, 2016.
- [20] Tao, Michael W., et al. "Depth from combining defocus and correspondence using light-field cameras." Proceedings of the IEEE International Conference on Computer Vision. 2013.
- [21] Mansour, Mostafa, et al. "Relative importance of binocular disparity and motion parallax for depth estimation: a computer vision approach." Remote Sensing 11.17 (2019): 1990.
- [22] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

UNSUPERVISED BLIND IMAGE QUALITY ASSESSMENT BASED ON MULTI-FEATURE FUSION

Qinglin He, Chao Yang and Ping An

School of Communication and Information Engineering,
Shanghai University, Shanghai, China

ABSTRACT

Image quality affects the visual experience of observers. How to accurately evaluate image quality has been widely studied by researchers. Unsupervised blind image quality assessment (BIQA) requires less prior knowledge than supervised ones. Besides, there is a trade-off between accuracy and complexity in most existing BIQA methods. In this paper, we propose an unsupervised BIQA framework that aims for both high accuracy and low complexity. To represent the image structure information, we employ Phase Congruency (PC) and gradient. After that, we calculate the mean subtracted and contrast normalized (MSCN) coefficient and the Karhunen-Loève transform (KLT) coefficient to represent the naturalness of the images. Finally, features extracted from both the pristine and the distorted images are adopted to calculate the image quality with Multivariate Gaussian (MVG) model. Experiments conducted on six IQA databases demonstrate that the proposed method achieves better performance than the state-of-the-art BIQA methods.

KEYWORDS

Blind Image Quality Assessment (BIQA), Unsupervised Method, Natural Scene Statistics (NSS), Karhunen-Loève Transform (KLT).

1. INTRODUCTION

With the rapid development of multimedia, the quality of images not only affects the visual experience of observers but also has an impact on image processing algorithms. How to measure image quality with lower computational complexity and better generalization performance has been a hot spot. The score of each image is the main evaluation criteria. There are two primary categories of score acquiring, namely the subjective evaluation and the objective evaluation. For the subjective evaluation, scores of different people on the same picture are needed and that is costly and time-consuming. On the other hand, models without human involvement are easy to use on large-scale databases for objective evaluation. The goal of image quality assessment (IQA) is to fit the objective score as close as possible to the subjective score, which means we could extract well-chosen features to imitate human behavior for more precise scores.

In general, the objective image quality assessment can be classified into three types which are full-reference (FR) IQA [1-6], reduced-reference (RR) IQA [7-10], and no-reference (NR) IQA [11-14]. FR IQA methods need the original image and its distorted version to fit the model, and RR IQA methods need features of the original image. NR IQA, which is also called Blind IQA (BIQA), only needs distorted images to predict scores. In FR and RR IQA methods, the need for

the original image as a reference limits the practical use. On the contrary, the BIQA methods do not have such strict requirements for model fitting and evaluation.

BIQA can be divided into supervised and unsupervised approaches. Supervised approaches usually utilize subjective scores as the ground truth to train the model. Mittal *et al.* [11] extracted natural scene statistics (NSS) features from local normalized images and Yang *et al.* [15] employed Karhunen-Loève transform (KLT) for learning-based features extraction, then these features were projected to subjective scores using support vector regression. Zeng *et al.* [16] used probabilistic quality representation and a more robust function for training the deep BIQA model. Ma *et al.* [17] proposed a multi-task learning-based deep learning approach, which consists of distortion identification and quality prediction tasks. Zhu *et al.* [18] proposed a deep meta-learning model for prior knowledge learning with good generalization ability. By simulating the human visual system, Chang *et al.* [19] used a visual neuron matrix (VNM) evaluator for quality assessment.

Unlike supervised approaches, unsupervised approaches can reveal better generalization capability with few manual calibration data. Wu *et al.* [13] proposed a highly efficient method for real-time evaluation. Wu *et al.* [12] proposed a visual perception nature image quality evaluation model for score training, which had an understanding-based global-local structure to simulate the top-down structure. Natural image quality evaluator (NIQE) [14] and its feature enriched extension, integrated local NIQE (ILNIQE) [20] introduced multivariate Gaussian (MVG) model for BIQA which required no subjective scores for regression model training. Liu *et al.* [21] introduced structure, naturalness, and perception features to the NIQE framework for further study.

In this paper, we propose a multi-feature fusion NIQE with better performance and lower complexity. We select Phase Congruency (PC) and gradient as structure features and select mean subtracted and contrast normalized (MSCN) and KLT coefficient as Natural Scene Statistics features. Fused structure features and NSS features are used for the MVG model fitting. Experiments show that the proposed unsupervised method achieves better performance with lower computational complexity on different databases. The rest of this paper is organized as follows. Section II introduces the detailed framework of our method. Section III reports the experimental results, and Section IV concludes this paper.

2. PROPOSED METHOD

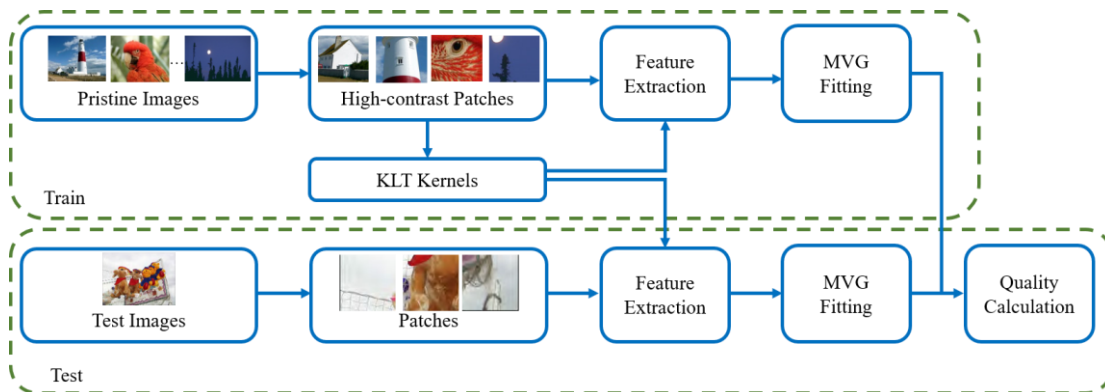


Figure 1. Framework of the proposed method

The framework of the proposed method is shown in Figure 1. To form the feature matrix, we extract structure features and NSS features from each non-overlapping patch of the image, which contains PC and gradient in the former, MSCN together with KLT coefficient in the latter. Then, we fit the MVG model with the feature matrix of pristine images as a benchmark. The distance between the benchmark model and the MVG model of the distorted image is taken as the objective score.

2.1. Structure features

2.1.1. Phase Congruency

Phase congruency calculates the maximum moment of PC covariance, which is used as an indicator of edge strength. We utilize [22] to compute the PC map of an image. For 2D signal s , the responses of even and odd-symmetric filters at position p can be denoted as

$$[e_{n,\theta_f}(p), o_{n,\theta_f}(p)], \quad (1)$$

where n and θ_f refer to scale and direction respectively, and $\theta_f = f\pi / F$, $f = 0, 1, \dots, F-1$ where F is the number of filter directions. The local amplitude is $A_{n,\theta_f}(p) = \sqrt{e_{n,\theta_f}(p)^2 + o_{n,\theta_f}(p)^2}$. Let $E_{n,\theta_f}(p) = \sum_n e_{n,\theta_f}(p)$, $O_{n,\theta_f}(p) = \sum_n o_{n,\theta_f}(p)$. Phase congruency is calculated using:

$$PC_{2D}(p) = \frac{\sum_f H_{\theta_f}(p)}{\varepsilon + \sum_f \sum_n A_{n,\theta_f}(p)} \quad (2)$$

where $H_{\theta_f}(p) = \sqrt{E_{n,\theta_f}(p)^2 + O_{n,\theta_f}(p)^2}$ and ε is a small positive constant.

We calculate the PC feature from the color relevant space O , which is converted from RGB in [23]:

$$\begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix} = \begin{bmatrix} 0.06 & 0.63 & 0.27 \\ 0.30 & 0.04 & -0.35 \\ 0.34 & -0.60 & 0.17 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3)$$

Finally, Weibull distribution is used to model the PC distribution in each color channel with scale parameter λ and shape parameter q . The dimension of the feature is 1×6 :

$$f(x; \lambda, q) = \begin{cases} \frac{q}{\lambda} \left(\frac{x}{\lambda} \right)^{q-1} \exp\left(-\left(\frac{x}{\lambda}\right)^q\right) & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (4)$$

2.1.2. Image Gradient

The gradient is an indispensable IQA index that represents the contrast and luminance information of an image. We use filters $D_h = [1, -1]$ and $D_v = [1, -1]^T$ to compute the horizontal and vertical gradient:

$$\begin{cases} G_h = I * D_h \\ G_v = I * D_v \end{cases} \quad (5)$$

where I refers to the image patch in the luminance channel and $*$ is the convolution operation. The distribution of G_h and G_v could be modeled as zero-mean General Gaussian Distribution (GGD) in [21]:

$$f(x; \alpha, \sigma^2) = \frac{\alpha}{2\beta(1/\alpha)} \exp\left(-\left(\frac{|x|}{\beta}\right)^\alpha\right) \quad (6)$$

where β refers to standard deviation $\beta = \sigma \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}$, and $\Gamma(\cdot)$ refers to gamma function, defined as:

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \quad a > 0 \quad (7)$$

We use α and σ as features and get a feature vector with a dimension of 1×4 .

2.2. NSS features

2.2.1. Mean Subtracted and Contrast Normalized Coefficient

Natural images and distorted images have different MSCN coefficient distributions. The extraction of MSCN coefficients from the image patch I in the luminance channel is as follows:

$$M(i, j) = \frac{I(i, j) - \mu(i, j)}{\sigma(i, j) + 1} \quad (8)$$

$$\mu(i, j) = \sum_{l=-3}^3 \sum_{k=-3}^3 \omega_{l,k} I(i+k, j+l) \quad (9)$$

$$\sigma^2(i, j) = \sum_{l=-3}^3 \sum_{k=-3}^3 \omega_{l,k} [I(i+k, j+l) - \mu(i, j)]^2 \quad (10)$$

where i and j indicate the coordinates of the pixel and $\{\omega = \omega_{l,k} \mid l = -3, \dots, 3, k = -3, \dots, 3\}$ defines a unit-volume Gaussian window.

GGD in Eq. (6) is used to fit MSCN distribution and get a 2-dimension feature. Furthermore, the asymmetric generalized Gaussian distribution (AGGD) model is applied to fit adjacent MSCN coefficients along with four directions, *i.e.* horizontal, vertical, main diagonal, and sub-diagonal [11]. Wherein the AGGD model is calculated:

$$f(x; \gamma, \beta_l, \beta_r) = \begin{cases} \frac{\gamma}{(\beta_l + \beta_r)\Gamma(1/\gamma)} \exp(-(\frac{-x}{\beta_l})^\gamma) \forall x \leq 0 \\ \frac{\gamma}{(\beta_l + \beta_r)\Gamma(1/\gamma)} \exp(-(\frac{-x}{\beta_r})^\gamma) \forall x \geq 0 \end{cases} \quad (11)$$

where γ controls the shape, β_l and β_r represent left and right side scale respectively. The mean value of this distribution is calculated as below:

$$\eta = (\beta_r - \beta_l) \frac{\Gamma(2/\gamma)}{\Gamma(1/\gamma)} \quad (12)$$

The parameters $(\gamma, \beta_l, \beta_r, \eta)$ for adjacent MSCN coefficient are set as features with a dimension of 1×16 .

2.2.2. Karhunen-Loève Transform

KLT is a data-driven feature extractor to extract image structural features [15]. Non-overlapping patches of MSCN normalized pristine image M with size $\sqrt{k} \times \sqrt{k}$ is used to collect vectorized patch r_u , wherein u is the patch index, and the covariance matrix C is defined as:

$$\begin{aligned} C &= E[(r - m)(r - m)^T] \\ &= \frac{1}{U} \sum_{u=1}^U [(r_u - m)(r_u - m)^T], \end{aligned} \quad (13)$$

where U is the number of training patches, and m is the average vector of each vectorized patch. Then, the KLT kernel with size $k \times k$ is the eigenvectors of the covariance matrix in Eq. (13), denoted as P . Each column vector in P is an eigenvector of the covariance matrix C , and these eigenvectors are arranged in descending order according to their eigenvalues.

GGD is chosen to fit the KLT coefficient distribution, the kernel size k is set to 4 and the feature dimension is 1×8 .

2.3. MVG fitting for Unsupervised BIQA

The features extracted above can be fitted with the MVG model as follows:

$$f(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp(-\frac{1}{2} (x - \nu)^T \Sigma^{-1} (x - \nu)) \quad (14)$$

where x represents the feature extracted from the image patches and d is the dimension of the feature vector, ν and Σ refer to the mean vector and covariance matrix of x respectively. In this paper, all features are extracted at two scales, *i.e.* the original image scale, and the down-sampled scale by a factor of 2. The corresponding image patch size of I is 96×96 and 48×48 . Therefore, the dimension of the features extracted from each image patch is 1×72 .

Then, the quality of the distorted image is measured as the distance between MVG parameters of the pristine images and distorted image:

$$Q = \sqrt{(\nu_1 - \nu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\nu_1 - \nu_2)} \quad (15)$$

where ν_1 , ν_2 and Σ_1 , Σ_2 are the mean vectors and covariance matrices of the pristine MVG model and the distorted image's MVG model.

3. EXPERIMENTS

3.1. Databases and Evaluation Methodology

Six widely utilized IQA databases including LIVE [25], MICT [26], CSIQ [27], TID2013 [28], CID2013 [29] and LIVE Challenge [30] *i.e.* LIVE-C are used to test the performance of the proposed method. We utilize the full LIVE and MICT databases for experiments. While for CSIQ and TID2013, we test on common distortion types, *i.e.* JPEG, JPEG2000, White Noise, and Gaussian Blur for a fair comparison. LIVE-C and CID2013 have real-world distortion without specific distortion types, therefore we test on the whole database respectively. We employ 125 images in [14] to train the KLT kernels and fit the pristine MVG model.

3.2. Overall Performance on 6 Databases

For the supervised models, we use the full LIVE database to train and then test the model on the rest five databases. For a fair comparison, we choose the three most commonly used criteria for model evaluation, which are Spearman Rank Order Correlation Coefficient (SROCC), Pearson Linear Correlation Coefficient (PLCC), and Root Mean Squared Error (RMSE). We calculate the SROCC with predicted scores and subjective scores, while for the calculation of PLCC and RMSE, we mapped the objective scores to the space of subjective scores with the nonlinear mapping method in [31].

The results of unsupervised methods on LIVE are in Table 1. The proposed method reaches the best results of three criteria. Table 2 shows the SROCC of the proposed method as well as other BIQA methods. “W. A.” refers to the weighted average performance over the five databases and the weights are the number of images selected in each database. The best performances of supervised and unsupervised methods are highlighted in bold. The generalization of RankIQA is pretty good among supervised models, while the weighted average performance of the proposed method is the highest among the unsupervised methods, even higher than RankIQA.

Table 1. The performance of unsupervised BIQA models on LIVE, which contains SROCC, PLCC, and RMSE.

method	SROCC	PLCC	RMSE
LPSI[13]	0.8181	0.8280	15.3184
NIQE[14]	0.9080	0.9064	11.5429
ILNIQE[20]	0.8972	0.9021	11.7913
SNP-NIQE[21]	0.9086	0.9073	11.4893
Proposed	0.9121	0.9095	11.3603

Table 2. SROCC results on Different Databases.

SROCC	MICT	CSIQ	TID2013	CID2013	LIVE-C	W.A.
BRISQUE[11]	0.8526	0.8842	0.8401	0.5485	0.3026	0.5866
MEON[17]	0.8919	0.9300	0.9012	0.3813	0.3640	0.6062
RankIQA[24]	0.9109	0.8337	0.8670	0.7040	0.3879	0.6437
LPSI[13]	0.9005	0.7711	0.7046	0.3230	0.0834	0.4180
NIQE[14]	0.8472	0.8711	0.7966	0.6568	0.4498	0.6528
ILNIQE[20]	0.7384	0.8794	0.8422	0.3057	0.4389	0.5946
SNP-NIQE[21]	0.8908	0.9024	0.8571	0.7155	0.4652	0.6879
Proposed	0.8745	0.9027	0.8764	0.7753	0.5036	0.7155

The SROCC results of 24-distortion-types on TID2013 are tabulated in Table 3 and the best results of each type are highlighted in bold. “Avg.” refers to the average score over the 24 distortions. Distortion types vary in the TID2013 database, so reaching the highest score on each type of distortion is a great challenge for models. LPSI, SNP-NIQE, and the proposed method all have six results in bold. Among them, the proposed method has the highest average score. Besides, the results of the proposed method are competitive on TID2013 for both common and uncommon distortion types.

Table 3. SROCC results on TID2013 in different distortions.

TID2013	1	2	3	4	5	6	7	8	9	10	11	12
LPSI	0.769 0	0.495 5	0.696 8	0.046 2	0.925 0	0.432 4	0.853 7	0.840 8	0.248 7	0.912 3	0.898 8	0.091 1
NIQE	0.814 8	0.590 6	0.541 1	0.721 1	0.851 0	0.744 7	0.860 8	0.809 7	0.577 6	0.859 7	0.866 0	0.121 6
ILNIQE	0.875 9	0.814 5	0.923 4	0.511 6	0.869 1	0.753 2	0.873 2	0.814 3	0.748 3	0.834 6	0.860 6	0.274 3
SNP-NIQE	0.885 6	0.733 0	0.649 5	0.740 0	0.873 0	0.799 7	0.857 3	0.863 8	0.612 8	0.879 1	0.877 6	0.281 7
Proposed	0.832 7	0.729 1	0.825 7	0.717 8	0.857 5	0.785 8	0.910 1	0.831 1	0.687 2	0.900 7	0.910 9	0.332 8
13	14	15	16	17	18	19	20	21	22	23	24	Avg.
0.6106	0.052 0	0.137 2	0.340 9	0.199 2	0.301 8	0.695 9	0.018 1	0.235 6	0.899 8	0.695 3	0.862 0	0.510 8
0.3997	0.033 1	0.174 1	0.147 0	0.106 3	0.302 1	0.678 8	0.053 5	0.754 2	0.760 3	0.568 3	0.851 7	0.549 5
0.5228	0.079 6	0.129 7	0.181 9	0.014 2	0.165 9	0.690 1	0.353 8	0.828 7	0.748 8	0.680 0	0.864 9	0.600 6
0.5917	0.014 9	0.032 1	0.099 9	0.156 2	0.106 0	0.740 1	0.208 3	0.830 0	0.790 0	0.634 7	0.828 7	0.586 9
0.4267	0.000 7	0.198 8	0.068 1	0.341 8	0.255 5	0.681 4	0.299 3	0.826 2	0.794 1	0.636 8	0.886 3	0.614 1

3.3. Ablation Test

To demonstrate the effectiveness of the structure and NSS features, we report the ablation test in Table 4. NSS features take the leading role while structure features play as a supplement. The combination of these two types of features can significantly improve the performance, both of these features are indispensable.

Table 4. Performance contribution of each type of feature and their combination of SROCC.

Database	structure features	NSS features	Proposed
LIVE	0.7137	0.9068	0.9121
MICT	0.6272	0.8713	0.8745
CSIQ	0.6100	0.8976	0.9027
TID2013	0.6063	0.8681	0.8764
CID2013	0.7106	0.7162	0.7753
LIVE-C	0.4420	0.4865	0.5036

3.4. Significance Test

To verify the statistical significance of the results, we applied t-test [25] on the prediction residuals of different objective methods.

Table 5. Statistical significance results between SROCC values. 1, 0, or -1 implies proposed method is statistical superior, comparative, or inferior to the algorithm with 95% confidence.

SROCC	LIVE	MICT	CSIQ	TID2013	CID2013	LIVE-C
BRISQUE	-	-1	-1	0	-1	-1
MEON	-	0	-1	0	1	1
RankIQA	-	0	0	0	1	0
LPSI	1	0	1	1	1	1
NIQE	0	0	0	0	1	0
ILNIQE	1	1	0	1	1	0
SNP-NIQE	0	0	0	1	1	0

‘1’, ‘0’ and ‘-1’ in Table 5 indicate that the proposed method is statistically superior, comparative, or inferior to the competing method on each database with 95% confidence. The unsupervised method is a little inadequate compared with supervised ones. The proposed method is no worse than other unsupervised methods. The proposed method has comparable performance with NIQE and SNP-NIQE, however, it is better than ILNIQE and LPSI on more than half of the databases.

3.5. Computation Complexity Comparison

Table 6 shows the average running time of different unsupervised BIQA methods on LIVE. These five methods are implemented on the MATLAB platform and tested on our PC with the following configuration, CPU: Intel Core i7-3770 3.40GHz Dual-Core, RAM: 8GB, and Windows system. All images in the LIVE database are utilized for the running time test. The generalization performance is good for ILNIQE, but it has higher computational complexity. The average running time of LPSI is very short, but the accuracy and generalization ability are limited. The proposed method has the highest results of six databases on a weighted average, the generalization performance and running time are competitive.

Table 6. Average running time of different unsupervised BIQA methods on LIVE.

	LPSI	NIQE	SNP-NIQE	ILNIQE	Proposed
Times(s)	0.02	0.24	5.06	5.49	1.21

4. CONCLUSION

In this paper, we propose an unsupervised BIQA method based on multi-feature fusion using structure features and NSS features. We extract PC, gradient, MSCN, and KLT features from non-overlapping image patches to fit the MVG feature matrix. The distance between the pristine and distorted MVG feature matrices is used as the objective score. Experiments on six IQA databases show that the proposed method achieves better performance with lower computation complexity on both common distortion types and real-world distortion. In the future, we can extend our work to uncommon distortion types.

ACKNOWLEDGEMENTS

This work was supported in part by the NSFC under Grant 61901252.

REFERENCES

- [1] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.
- [2] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, 2003, pp. 1398-1402 Vol.2, doi: 10.1109/ACSSC.2003.1292216.
- [3] L. Zhang, L. Zhang, X. Mou and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment," in *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378-2386, Aug. 2011, doi: 10.1109/TIP.2011.2109730.
- [4] J. Farah, M. Hojeij, J. Chrabieh and F. Dufaux, "Full-reference and reduced-reference quality metrics based on SIFT," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 161-165, doi: 10.1109/ICASSP.2014.6853578.
- [5] X. Ma, X. Jiang and X. Guo, "Full-reference image quality assessment based on the analysis of distortion process," *2017 4th International Conference on Systems and Informatics (ICSAI)*, 2017, pp. 1256-1260, doi: 10.1109/ICSAI.2017.8248471.
- [6] J. Kim and S. Lee, "Deep blind image quality assessment by employing FR-IQA," *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3180-3184, doi: 10.1109/ICIP.2017.8296869.
- [7] R. Soundararajan and A. C. Bovik, "RRED indices: Reduced reference entropic differencing framework for image quality assessment," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1149-1152, doi: 10.1109/ICASSP.2011.5946612.
- [8] J. Wu, Y. Liu, G. Shi and W. Lin, "Saliency change based reduced reference image quality assessment," *2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1-4, doi: 10.1109/VCIP.2017.8305041.
- [9] Y. Liu, G. Zhai, K. Gu, X. Liu, D. Zhao and W. Gao, "Reduced-Reference Image Quality Assessment in Free-Energy Principle and Sparse Representation," in *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 379-391, Feb. 2018, doi: 10.1109/TMM.2017.2729020.
- [10] Q. Hu, Y. Sheng, L. Yang, Q. Li and L. Chai, "Reduced-Reference Image Quality Assessment for Single-Image Super-Resolution Based on Wavelet Domain," *2019 Chinese Control and Decision Conference (CCDC)*, 2019, pp. 2067-2071, doi: 10.1109/CCDC.2019.8833247.
- [11] A. Mittal, A. K. Moorthy and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," in *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695-4708, Dec. 2012, doi: 10.1109/TIP.2012.2214050.
- [12] L. Wu, X. Zhang, H. Chen, D. Wang, and J. Deng, "Vp-niqe: An opinion-unaware visual perception natural image quality evaluator," *Neurocomputing*, vol. 463, pp. 17-28, 2021.
- [13] Q. Wu, Z. Wang, and H. Li, "A highly efficient method for blind image quality assessment," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 339-343.
- [14] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'Completely Blind' Image Quality Analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209-212, 2013.

- [15] C. Yang, X. Zhang, P. An, L. Shen, and C. J. Kuo, "Blind image quality assessment based on multi-scale klt," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.Z.
- [16] H. Zeng, L. Zhang, and A. C. Bovik, "Blind image quality assessment with a probabilistic quality representation," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 609–61.
- [17] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo, "End-to-end blind image quality assessment using deep neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1202–1213, 2018.
- [18] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Metaiqa: Deep meta-learning for no-reference image quality assessment," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 131–14 140.
- [19] H.-W. Chang, X.-D. Bi, and C. Kai, "Blind image quality assessment by visual neuron matrix," *IEEE Signal Processing Letters*, vol. 28, pp. 1803–1807, 2021.
- [20] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2579–2591, 2015.
- [21] Y. Liu, K. Gu, Y. Zhang, X. Li, G. Zhai, D. Zhao, and W. Gao, "Unsupervised blind image quality evaluation via statistical measurements of structure, naturalness, and perception," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 929–943, 2020.
- [22] P. Kovess, "Image features from phase congruency," *Videre J. Comput. Vision Res.*, vol. 1, 01 1999.
- [23] J. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts, "Color invariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1338–1350, 2001.
- [24] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Rankiqa: Learning from rankings for no-reference image quality assessment," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1040–1049.
- [25] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, 2006.
- [26] Y. Horita, K. Shibata, and Y. Kawayoka, "Toyama image quality evaluation database," 2011, <http://mict.eng.u-toyama.ac.jp/mictdb.html>.
- [27] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *Journal of Electronic Imaging*, vol. 19, no. 1, p. 011006, 2010.
- [28] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C. J. Kuo, "Color image database tid2013: Peculiarities and preliminary results," in *European Workshop on Visual Information Processing (EUVIP)*, 2013, pp. 106–111.
- [29] T. Virtanen, M. Nuutinen, M. Vaahteranoksa, P. Oittinen, and J. Häkkinen, "CID2013: A Database for Evaluating No-Reference Image Quality Assessment Algorithms," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 390–402, 2015.
- [30] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 372–387, 2016.
- [31] Antkowiak, J., Baina, T.J., 2000. Final report from the video quality experts group on the validation of objective models of video quality assessment march. ITU-T Standards Contribution COM.

AUTHORS

Qinglin He received the B.E. degree in communication system from the School of Communication and Information Engineering, Shanghai University, Shanghai, China, in 2019. She is currently pursuing the M.E. degree in Information and Communication Engineering from Shanghai University. Her research interests include computer vision and image quality assessment.



Chao Yang received the B.E. and Ph.D. degree from the School of Communication and Information Engineering, Shanghai University, Shanghai, China, in 2012 and 2017, respectively. From Nov. 2017 to Oct. 2018, he was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Southern California (USC), Los Angeles, CA, USA. He joined the School of Communication and Information Engineering, Shanghai University, Shanghai, in 2019, where he is currently a Lecturer. His current research interests include video processing, video compression, and image quality assessment.



Ping An received the B.E. and M.E. degrees from the Hefei University of Technology, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree from Shanghai University, Shanghai, China, in 2002. In 1993, she joined Shanghai University, where she is currently a Professor with the Video Processing Group, School of Communication and Information Engineering. From 2011 to 2012, she joined the Communication Systems Group, Technische University at Berlin, Germany, as a Visiting Professor. She has finished over 15 projects supported by the National Natural Science Foundation of China, the National Science and Technology Ministry, and the Science and Technology Commission of Shanghai Municipality. Her research interests include image and video processing, with a focus on immersive video processing. She was a recipient of the Second Prize of the Shanghai Municipal Science and Technology Progress Award in 2011, the Second Prize in Natural Sciences of the Ministry of Education in 2016, and the Second Prize in Natural Sciences of the Chinese Institute of Electronics in 2018.



MEETING CHALLENGES OF MODERN STANDARD ARABIC AND SAUDI DIALECT IDENTIFICATION

Yahya Aseri, Khalid Alreemy, Salem Alelyani, Mohamed Mohana

Center for Artificial Intelligence, King Khalid University, Saudi Arabia

ABSTRACT

Dialect identification is a prior requirement for learning lexical and morphological knowledge a language variation that can be beneficial for natural language processing (NLP) and potential AI downstream tasks. In this paper, we present the first work on sentence-level Modern Standard Arabic (MSA) and Saudi Dialect (SD) identification where we trained and tested three classifiers (Logistic regression, Multi-nominal Naïve Bayes, and Support Vector Machine) on datasets collected from Saudi Twitter and automatically labeled as (MSA) or SD. The model for each configuration was built using two levels of language models, i.e., unigram and bi-gram, as feature sets for training the systems. The model reported high-accuracy performance using 10-fold cross- validations with average 98.98%. This model was evaluated on another unseen, manually-annotated dataset. The best performance of these classifiers was achieved by Multi-nominal Naïve Bayes, reporting 89%

KEYWORDS

Dialect Identification, NLP, Standard Arabic, Saudi Dialect, Classification.

1. INTRODUCTION

Human language understanding and generation is an essential component for developing numerous AI systems, such as virtual assistants, chatbots, talking robots. This component requires lexical resources, morphological and grammatical knowledge, and meaning representation that captures users' intents and facilitates human-machine interaction, in particular conversational interface, in an efficient and powerful way. Though Arabic natural language processing, as tools for human- machine interaction, has received considerable attention, the differences between spoken/dialectal and standard Arabic pose challenges to natural language processing and potential AI applications [1]. The NLP tools developed for Modern Standard Arabic (MSA) often fail in dealing with modern varieties of Arabic. Dialectal identification (DI), however, is considered an important NLP task that can be utilized for developing lexical resources and prepossessing large-scale data used for machine learning tasks.

Dialectal identification is a form of Language Identification (LI), which is the task of detecting the natural language that a document or part thereof is written in so that a system can mimic the human ability of recognizing certain languages [2]. Researchers in this area do not make a distinction between languages and language varieties/dialects since the computational methods used are identical and challenges faced are similar. Furthermore, the motivation for LI or DI is also almost the same. Though LI was initially motivated by machine translation, it is considered a fundamental component for natural language processing of languages with a high degree of dialectal variety. To ensure that a given document is relevant to NLP tools available, LI is used to

determine the language of the document and whether it is subject to further natural language processing. Moreover, LI plays a vital role in creating lexical resources and corpora used in machine learning tasks, in particular for low- resource languages or dialects.

A major challenge, for Arabic NLP, comes from the fact that Arabic language exists in a state of diglossia [3] in which the standard form of the language, MSA, and the regional dialects live side-by-side and are closely related [4]. While MSA refers to the language used in Arab world used in education, newspapers, and laws documentation, Dialectal Arabic (DA) refers to spoken language (or informal written language) used in daily communication. Spoken Arabic exhibits several language variations, which are a mixture of MSA and a number of Arabic vernaculars. These language variations are what people in Arab world acquire and speak at home and use in their daily lives. From a natural language processing perspective, there are five major groups of dialects that are regionally defined: Egyptian, Gulf, Iraqi, Levantine, and Maghrebi [1, 5]. These variations differ from one another in terms of lexical, morphological and syntactic structures, though they have the core grammar in common. In addition, each group shows internal variations that cannot be neglected. For example, Gulf dialects include Bahraini, Kuwaiti, Omani, Qatari, Saudi, UAE, and Yamani [5]. As we will see in section 2, the aforementioned linguistic diversity has been taken into consideration by the Arabic NLP community and two machine learning models have been proposed. One is a multi-way classification model where the classes range from 3 to 29 classes. The other model is binary-classification model where the task is to distinguish a certain dialect from MSA.

2. RELATED WORK

Arabic dialects identification has recently attracted the Arabic NLP researchers and practitioners [6, 7]. Studies presented, however, differ in terms of their aims, target dialects, and approaches. Some focus on systems performing binary classification between MSA and a specific dialect [4, 8, 9], others describe multi-way classifications between (MSA) and other dialects, including the five major dialects: Egyptian, Gulf, Iraqi, Levantine, and Maghrebi [9– 18]. These studies have implemented different methods of traditional machine learning and deep learning algorithms [2].

For multi-dialectal identification, Harrat et al. [11] describe experiments using datasets representing Maghrebi dialects and what they call Middle Eastern dialects as well as MSA. Shervin Malmasi et al. [14] present work on sentence-level Arabic dialects identification. Using a set of surface character and word features, they trained their system on a multidialectal parallel corpus of Arabic. This work shows 74% accuracy on a 6-way multi-dialect classification. Mohamed Lichouri et al. [12] describe methods for textual Arabic dialects identification. The experiments were conducted on two datasets: one represents Maghrebi and Middle Eastern dialects, while the other represents Algerian dialects. For the Middle Eastern dialects, the system achieved an average accuracy of 92% and 76% for Algerian dialects. Leena Lulu et al [15] describe deep learning models used for the automatic classification of Arabic dialectal texts. They used the Arabic Online Commentary (AOC), which includes Egyptian (EGP), and Gulf (GLF), and Levantine (LEV) dialects. Mohamed Ali [16] introduces systems submitted to the Arabic dialect identification shared task 2018, which included MSA, Egyptian, Gulf, Levantine, and north African dialects. For this task, he used character-level convolution neural network as well as dialect embedding vectors, achieving 57.6% F1-score. Mohamed Elaraby et al. [18] used the AOC for both binary and multi-way classification. Having benchmarked the data, they trained and tested six different deep learning methods and compared the results to several classical machine learning models, showing 87.65% accuracy on the binary task (MSA vs. dialects), 87.4% on the three-way dialect task (Egyptian vs. Gulf vs. Levantine), and 82.45% on the four-way variants task (Egyptian vs. Gulf vs. Levantine vs. MSA).

In an attempt to provide a fine-grained classification, Sadat et al. [10] present work on 18 local Arabic dialects. To develop probabilistic models, they used the character n-gram Markov language model and Naive Bayes classifiers trained on datasets derived from social media. Abdul Mageed et al. [19] also describe work for detecting dialects from 29 cities in 10 Arab countries. Similarly, Mohammad Salameh et al. [20] developed a fine-grained system with 25-way classification where the labels are 25 cities from several countries (including Riyadh and Jeddah in Saudi Arabia) as well as MSA. Their systems were trained to predict the location/city of the speaker rather than to give linguistic labels, i.e., dialectal classes, to a given text. For binary classification tasks, Elfardy et al. [4, 8, 9] introduce a system performing binary classification between EGP and MSA. Elfardy et al. [4] present work for sentence-level binary classification task performed on EGP and MSA. They implemented supervised machine learning algorithms to train their system, using token level features and other meta features, to predict the correct label for a given sentence. The system achieved an accuracy of 85.5% on the AOC dataset. Tillmann et al. [8] present another work to perform the same task, i.e., classification between EGP Arabic and MSA. The system was also tested on the AOC dataset and achieved an accuracy of 89.1 %. However, they indicate that the system's performance decreased when evaluation on data from another source. Al-Badrashiny et al. [9] also focus on MSA and EGP. However, they describe a hybrid approach in which a sentence-level classifier was trained to predict the correct class for each sentence using labels and the confidence scores generated by two underlying classifiers. Their system achieved an accuracy of 90.8%.

The focus of this paper is on Saudi dialect (SD) identification. To the best of our knowledge, this work presents the first identification system of SD. To avoid the over-fitting or under-fitting problems that may result from the linguistic differences among Arabic dialects, we adopt the binary classification model and introduce a system that performs binary classification between MSA and SD. Such a system takes Saudi Twitter texts as inputs and provides linguistic labels for each as either MSA or SD. Because there is a considerable overlap between Arabic dialects and MSA in terms of lexical, morphological, and syntactic properties, we define SD, in the present study, as any text that contains at least one token that lexically or morphologically belongs to the dictionary of SD defined in section 4.2.

3. RESEARCH PROBLEM AND DATA

In this paper, we present a system that discriminates between SD and MSA, which is to our knowledge the first work aiming at this goal. This is a significant step toward building a large lexicon and NLP tools used for AI systems that can understand this dialect.

Given the fact that dealing with spoken forms of Arabic language is not an easy task, Twitter is considered a good source for achieving this goal for one main reason. Twitter contains informal texts that are to a large extent close to spoken language in terms of the lexicon used in this dialect and morphological variations. However, Twitter also contains linguistic data that represent MSAs as well as data with code switching from MSA to SD and vice versa. To make use of Twitter in learning this dialect, it is important to first distinguish what can be pure MSA and what represents SD. Hence, this paper presents a system performing binary classification, which takes a sentence as an input and labels it either MSA or SD. SD, in this paper, is linguistically defined based on distinctive morphological properties (i.e., inflectional features) or lexical items (i.e., functional words) used in spoken language. Consequently, texts classified as SD are those containing code-switching between MSA and SD.

4. EXPERIMENTAL SETUP

We conducted the following experiment using three supervised machine learning algorithms: logistic regression (LR), multi-nominal Naïve Bayes (MNB), and support vector machine (SVM). These models were trained on 346,931 tweets collected from Masfah, an online platform, which has a huge number of tweets documents stored by crawling Twitter social network. The algorithms were trained on two subsets of data representing the two linguistic classes: MSA and SD. To train our model to identify SD and distinguish it from MSA texts, we need large-scale training data that represent these two classes. Because manual annotation is time-consuming and requires too much effort, we prepared our training data by automatic extracting and labeling tweets representing each class. The following sub-sections explain text preprocessing, building dictionaries, automatic labeling, model training and evaluation.

4.1. Text Preprocessing

Twitter posts are actually noisy data. Tweets are intended to contain texts, but also contain a mixture of other data types like images and videos. The cleaning process of the data includes the following tasks:

- Removing images, videos, hashtags, user mentions, symbolic characters, emojis, numeric characters, and hyperlinks.
- Elimination of Arabic diacritics *Tashkeel*.
- Replacing *Alif Hamza* (أ) with *plain Alif* (ا).
- Replacing *Taa Marbootah* (ة) with *Haa* (ه).
- Removing stop words.
- Deleting duplicate documents.

The removal of stopwords needs to be done carefully. Not all stopwords should be eliminated. Stopwords like other tokens can be true identifiers and hence many stopwords belong to what we call identification terms that distinguish the MSA sentences from SD sentences. To only eliminate stopwords list that is useless, we used Count Vectorizer for bag-of-words representation rather than TF-IDF. Moreover, common stopwords that appear in both MSA sentences and SD sentences were eliminated because they have no effect on our model. We also eliminated duplicate documents resulting from the removal of hashtags, user mentions, hyperlinks, etc. Such duplicate documents came out to be a problem since they have no value, which results in a negative effect on model training.

4.2. Building Dictionaries

Prior to building lexicons and preparing our training corpus, we had to make a few linguistic assumptions. First, we assume that MSA and SD represent two language levels that lexically and morphologically overlap. Secondly, each level may have its unique lexicon and grammatical properties. Thirdly, speakers of SD often use code switching from SD and MSA and vice versa. Taking these assumptions into consideration, we started with a short list of vocabulary that uniquely identify SD. This list was identified based on three types of features: lexical features, functional words, and inflectional morphology. Likewise, we came up with another short list of vocabulary that uniquely identify MSA and commonly used in formal writing. The two lists were revised and approved by linguists who are experts in both MSA and SD. Examples are listed in Table 1. The list of Vocabulary in each domain cannot belong to both domains at the same time. That is, each word is considered to belong to either MSA or SD. For example, the word “ل إ” is considered an SD identifier as it is widely used in spoken SD but cannot appear in MSA texts. On

the other hand, “فام” is a functional word that belongs to MSA; and thus, it is considered an MSA identifier that cannot be an SD identifier at the same time. We used these two short lists to extract text/sentences and group them into two sets of documents: MSA documents and SD documents. To expand the vocabulary lists that identify each class, we extracted all words from all texts in each document set and added them to the corresponding list, which create two dictionaries. The texts belonging to SD document set undoubtedly contain words that belong to MSA list, according to the third assumption previously stated. Thus, we removed the overlap section by eliminating the intersection between the two dictionaries. This process has resulted in a dictionary of pure MSA that contains 41861 words and another one that is pure SD with 51286 words.

4.3. Automatic Annotation

We used the updated dictionaries to re-filter the two class of documents: MSA documents and SD documents. The outputs are two classes that overlap because the informal texts/dialect contain MSA words. Figure 1 represents such an overlap between the two class of documents.

Documents lying in the overlapping region are either MSA documents that have SD identification words or SD documents that includes MSA identification words. Therefore, we eliminated these documents from our training datasets, which decreases the size of the overlap documents. This process of reducing the overlap size yielded a corpus that contains two datasets representing MSA (107,917 sentences) and SD (128,051 sentences). Sentences/documents of each class were automatically labeled either MSA or SD. The two datasets were combined and shuffled to ensure they were normally distributed and to avoid model overfitting.

Table 1: Examples of Tokens used as identifiers of MSA and SD

Standard Token			Dialect Token		
كيفما	ذهبت	استيقظ	بشويس	اهلين	ابغا
يؤدي	ربما	التي	بعدين	ايش	ابغى
لذلك	ريثما	الذي	بياخذ	بالمره	اركد
لقد	سوف	الذين	بيطلع	بايخ	اروح
لكي	سيقوم	حسبما	تبون	برضه	اشلون
من أجل	عندما	حيث	تبي	برضو	اشوف
منذ	فقد	حيثما	على راسي	بروح	الحين
هكذا	كما	ذهبتنا	ترا	يس	اللي

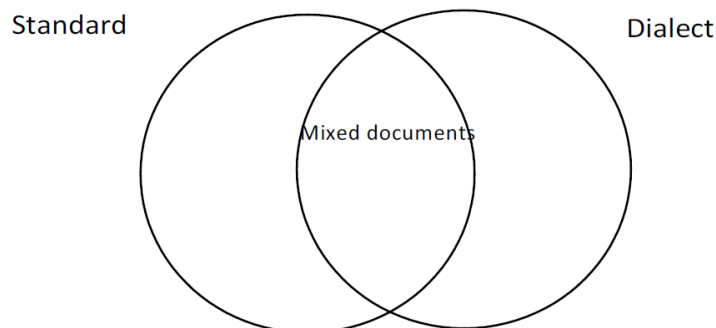


Figure 1: Documents Overlap

4.4. Model Training

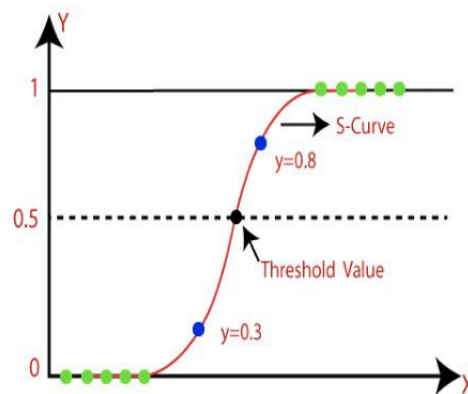
Table 2: Examples of Bi-gram of SD Expression Formed from two MSA Words

Dialect Term	Standard Word-2	Standard Word-1
زي الناس	← الناس	زي
يعطيك العافية	← العافية	يعطيك
على طول	← طول	على

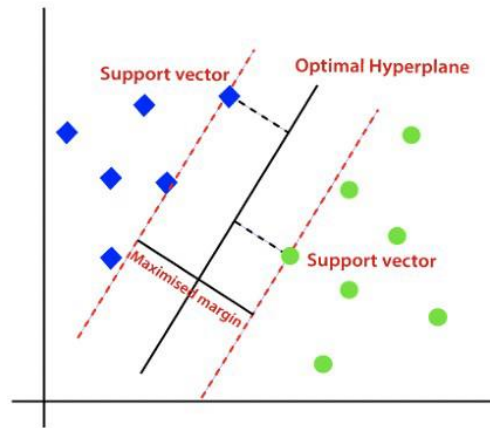
To train our models, each document/sentence was converted into vectors representation using CountVectorizer. The minimum document frequency was set to 10 to emphasize the effectiveness of the selected features. Features were extracted as n-gram terms where n was set to (1,2). We used bi-gram because in many cases an MSA word when combined with another MSA word form an SD phrase/expression. Examples of this phenomenon are shown in Table 2. We implemented three supervised machine learning algorithms to predict discrete values of (0 and 1) as MSA or SD respectively. We used Multi-nominal Naïve Bayes (MNB) classifier, which depends on Bayesian theorem described by the following formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Despite its simplicity, it in many cases outperforms other classification algorithms, as we see in our case. Thus, it is widely used in text classification. We also used Logistic Regression (LR). As shown in Figure 2, LR depends on the notion of probability and uses sigmoid function to convert continuous values into discrete numbers, which fits binary classification problems. In our case, LR assigns a probability that a given text belongs to a certain class. Finally, we implemented Support Vector Machine (SVM), an algorithm that can be used for both regression and classification problems. We used SVM because it can handle data with large features, as in our case, in which every data point is plotted in an n dimensional space. Such a classification task is done by finding the hyperplane that separates these spaces. The maximum distance between the nearest data points in every separated spaces is called margin, as shown in Figure 2.



a) Logistic Regression



b) Support Vector Machine

Figure 2: Difference between LR and SVM

Table 3: Accuracy Average of 10-fold Cross-validation

Algorithm	MNB	LR	SVM
Accuracy Average	98.24	99.44	99.28

The classifiers take n-gram of an input sentence and compute the probability/likelihood that this sentence belongs to the MSA class or the SD class. Instead of splitting the data into two subsets, we used k-fold cross-validation where k=10. The dataset was split into 10-folds and each model was repeatedly built using 90% for training while holding 10% for testing. For each model configuration, the accuracy of each fold was captured. Table 3 shows that the three models perform with a slight difference. The 10-fold average score for MNB, LR, and SVM is also reported as 98.24%, 99.44% and 99.28% respectively.

5. MODEL EVALUATION AND DISCUSSION

To evaluate the models' performance, the three classifiers were tested on an unseen, manually doubled-annotated dataset. The point here is to compare the model performance on this dataset to its performance on the dataset that was automatically labeled. The total number of this testset is 15747 Saudi tweets. Unlike training data that were automatically collected and labeled, this dataset was randomly collected from Saudi Twitter and manually annotated by language experts. It was cleaned and preprocessed, using the same tools mentioned earlier, and given to annotators who label each sentence as either MSA or SD. Figure 4 shows the performance at a probability threshold of 0.5. True Positive Rate (TPR) refers to the ratio of correctly predicted positive labels from all the positive labels, which is the MSA in our case, while False Positive Rate (FPR) refers to the ratio of incorrectly predicted positive labels from all the negative labels which is the SD. We computed their values as follows.

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

As we had 107,917 records for MSA and 128,051 records for SD in our training data, the percentages of MSA and SD with respect to the total records are 46% and 54% respectively. Such data can be considered balanced data. The predicted values for the validation data are considered either “0” for MSA or “1” for SD. By using the threshold value of 0.5, the prediction is considered as “0” when the predicted probability is in the range [0.0 - 0.49], and “1” when the prediction is in the range [0.5 – 1.0]. In contrast with the models’ performance reported above, Table 4 shows the performance these classifiers where the best result was achieved by by MNB classifier, reporting 89.05% for the model accuracy and 88% for F- score.

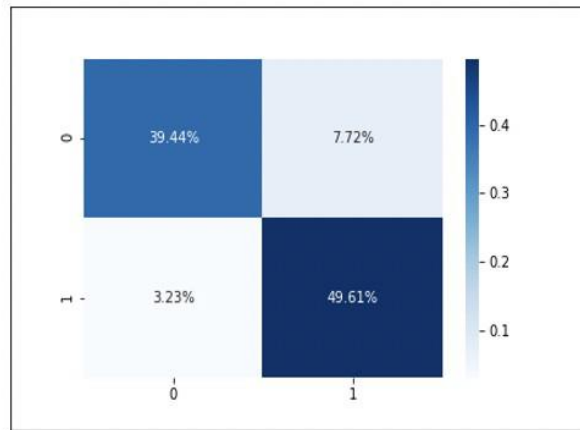


Figure 3: Confusion Matrix

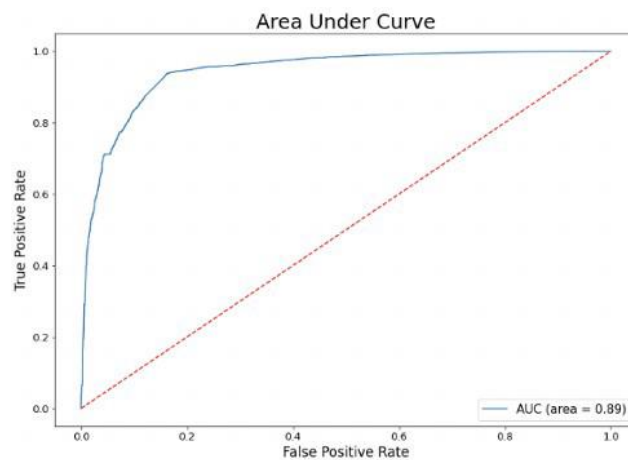


Figure 4: Area Under the Curve

Table 4: Models’ Performance on Double-Annotated Data

	Accuracy	Precision	Recall	F-score
Multinomial Naïve Bayes	89.05	0.92	0.84	0.88
Logistic Regression	63.31	0.94	0.24	0.38
Support Vector Machine	63.41	0.90	0.25	0.39

This experiment shows that MNB performs well on SD identification, though it is a difficult task because the high degree of similarity between MSA and SD and studies have shown the difficulty of language/dialect identification task performed on neighboring dialects or similar

languages [2, 11, 12, 14]. The drop in the result may be attributed to the fact that in SD a sentence that does not contain any dialectal token can also be recognized by human annotators as SD when it has a combination of two or three MSA words that form a SD phrase, as discussed previously. This linguistic phenomenon cannot be recognized by the system unless we have a large lexicon of n-gram (where $n > 1$) of SD, which is not available yet. In addition, named entities represent a challenge to any language or dialect identification task. In this experiment, these entities have been treated in our language model as regular tokens. We may also consider text normalization as another issue that needs to be taken care of in such a task. We believe that deep and careful text normalization is still required prior to training our model.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented work on sentence-level Saudi Dialect (SD) identification task where we trained and tested three classifiers on datasets collected from Saudi Twitter. Given that Saudi tweets represent two linguistic classes: MSA and SD, the task was to discriminate between Saudi dialect SD and MSA using supervised machine learning algorithms. We trained three classifiers: Logistic regression (LR), multi-nominal Naïve Bayes (MNB), and support vector machine (SVM) on a dataset that was automatically labeled as MSA or SD. The model for each configuration was built using two levels of language models (un-gram and bi-gram), as features for training. The systems reported high-accuracy performance (average 98.98%) when they were tested. However, when we tested these classifiers on another manually-annotated dataset and compared their results to automatic annotation, the best performance was reported by MNB achieving accuracy of 89.05. The drop in the performance probably occurred as a result of the factors mentioned previously in the discussion.

These results can be considered a baseline for future work. We look for improving our model by using additional feature sets and higher levels of language models. Moreover, we may use orthographic normalization tools that may have positive impacts on our model. As indicated, named entities represent another challenge for dialects identification. Hence, using NER tools can also be powerful for such a task. We seek to improve our systems knowing that SD identification is a significant step for learning lexical and morphological knowledge of this dialect, which can be beneficial for further Arabic NLP downstream tasks.

ACKNOWLEDGEMENT

The authors are grateful for the financial support received from King Khalid University for this research Under Grant No. R.G.P2/100/41.

REFERENCES

- [1] Omar F Zaidan and Chris Callison-Burch. "Arabic dialect identification". *Computational Linguistics* 40, pp. 171–202, 2014.
- [2] Tommi Jauhiainen et al. "Automatic language identification in texts: A survey". *Journal of Artificial Intelligence Research* 65, pp. 675–782, 2019.
- [3] Charles A Ferguson. "Diglossia". *word* 15, pp. 325–340, 1959.
- [4] Heba Elfardy and Mona Diab. "Sentence level dialect identification in Arabic". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume2: Short Papers)*. 2013. Pp. 456–461.
- [5] Abdulhadi Shoufan and Sumaya Alameri. "Natural language processing for dialectal Arabic: A Survey". In: *Proceedings of the second workshop on Arabic natural language processing*. 2015. Pp. 36–48.
- [6] Imane Guellil et al. "Arabic natural language processing: An overview". *Journal of King Saud University-Computer and Information Sciences*, 2019.

- [7] Kareem Darwish et al. "A panoramic survey of natural language processing in the Arab world". *Communications of the ACM* 64, pp. 72–81, 2021.
- [8] Christoph Tillmann, Saab Mansour, and Yaser Al-Onaizan. "Improved sentence-level arabic dialect classification". In: *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*. 2014. Pp. 110–119.
- [9] Mohamed Al-Badrashiny, Heba Elfardy, and Mona Diab. "Aida2: A hybrid approach for token and sentence level dialect identification in arabic". In: *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. 2015. Pp. 42–51.
- [10] Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. "Automatic identification of arabic dialects in social media". In: *Proceedings of the first international workshop on Social media retrieval and analysis*. 2014. Pp. 35–40.
- [11] Salima Harrat et al. "Cross-dialectal arabic processing". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2015. Pp. 620–632.
- [12] Mohamed Lichouri et al. "Word-Level vs Sentence-Level Language Identification: Application to Algerian and Arabic Dialects". *Procedia Computer Science* 142, pp. 246–253, 2018.
- [13] Shervin Malmasi et al. "Discriminating between similar languages and arabic dialect identification: A report on the third dsl shared task". In: *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*. 2016. Pp. 1–14.
- [14] Shervin Malmasi, Eshrag Refaee, and Mark Dras. "Arabic dialect identification using a parallel multidialectal corpus". In: *Conference of the Pacific Association for Computational Linguistics*. Springer. 2015. Pp. 35–53.
- [15] Leena Lulu and Ashraf Elnagar. "Automatic Arabic dialect classification using deep learning models". *Procedia computer science* 142, pp. 262–269, 2018.
- [16] Mohamed Ali. "Character level convolutional neural network for Arabic dialect identification". In: *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. 2018. Pp. 122–127.
- [17] Faisal Alshargi et al. "Morphologically Annotated Corpora for Seven Arabic Dialects: Taizi, Sanaani, Najdi, Jordanian, Syrian, Iraqi and Moroccan". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019. Pp. 137–147.
- [18] Mohamed Elaraby and Muhammad Abdul-Mageed. "Deep models for arabic dialect identification on benchmarked data". In: *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. 2018. Pp. 263–274.
- [19] Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. "You tweet what you speak: A city-level dataset of arabic dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.
- [20] Mohammad Salameh, Houda Bouamor, and Nizar Habash. "Fine-grained arabic dialect identification". In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018. Pp. 1332–1344.

AUTHORS

YAHYA ASERI is an assistant professor in Linguistics and Human language Technology in Arabic Language department at King Khalid University. He serves as a consultant at Center for Artificial intelligence at KKU University, and he is also a member of ACL SIGARAB (ACL Special Interest Group on Arabic Natural Language Processing). He obtained his PhD degree in Linguistics from University of Colorado-Boulder, USA. His research interests focus on theoretical computational linguistics and its applications to human language technology; namely, developing linguistic models for human language understanding, building annotated resources /corpora for machine learning, and developing natural language processing applications.

KHHALID ALREEMY is a software engineer with experiences in Artificial Intelligence, natural language processing and computer vision using deep learning models. He has a long experience working on C# for Microsoft and NET applications. He worked as a SQL database designer, specifically Oracle and MySQL exploiting the power of Python programming language for handling various data types. I also worked for diverse AI applications such as spam detection, sentiment analysis, object detection, Chatbots, and abnormality detection in mammogram images, etc. Currently he is working on various projects of Artificial Intelligence with highly oriented skills for data processing and preparation.

SALEM ALELYANI has been an assistant professor in the Computer Science Department at King Khalid University, Saudi Arabia, since 2014. He serves as a consultant to the president of the university and the director of the Center for Artificial Intelligence. He obtained his Ph.D. from Arizona State University in 2013 in Machine Learning and Data Mining. He has several publications in the field. He serves as a reviewer in multiple international conferences and scientific journals including ICTAI, AAAI, ICMLA, ICML, IRI, IEEE Access, Artificial Intelligence Review, AJSE, Information Sciences, and others. Also, he serves as a board member and as a PC member in other international journals and conferences.

MOHAMED MOHANA is a Mechatronics Engineer with a solid background in Control, Electronics, Mechanical, and Computer systems. He has done his Master's Engineering in Control and Automation using Artificial Intelligence; besides, He won the best master research from IEEE Malaysia Control System. He has four years of experience in Computer Vision and IoT, as well as MATLAB and Python programming languages. He has been involved in real-life industrial problems and overcame the challenges using IoT and Robots controlled by AI. He has created a UAV with an autopilot controller and a state-of-art IoT and Computer Vision device for potent security purposes. Recently, he is working with AI in renewable energy. However, he has the ability to see the whole picture from solving the problem (Research) up to the solution deployment (Development), to create Artificial Intelligence solutions and products for real-life situations.

AUTHOR INDEX

<i>Ahmed Bounekkar</i>	27
<i>Anand Rao</i>	197
<i>Ananth Kamath</i>	17
<i>Antoinette Young</i>	197
<i>Bo Lang</i>	237
<i>Chao Yang</i>	329
<i>Chidambaram Baskaran</i>	47
<i>Chun-Hsien Lin</i>	145
<i>Colin Hanley</i>	109
<i>David A. Noever</i>	97
<i>Diane Fulton</i>	127
<i>Dodo Khan</i>	225
<i>Evan Gunnell</i>	317
<i>Feng Dong</i>	109
<i>Francesco Bertolotti</i>	207
<i>Hao Zheng</i>	317
<i>Hazirah Bee Yusof Ali</i>	287
<i>Heng Zhou</i>	273
<i>Hongyu Liu</i>	237
<i>Hristo Petkov</i>	109
<i>Iftikhar U. Sikder</i>	121
<i>James J. Ribero</i>	121
<i>Jayakrishna Guddeti</i>	17
<i>Karuppiah Aravindhan</i>	47
<i>Khalid Alreemy</i>	341
<i>Kunhao Li</i>	237
<i>Lea Matlekovic</i>	253
<i>Lili Marziana Abdullah</i>	287
<i>Linlin Zhang</i>	91
<i>Low Tan Jung</i>	225
<i>Magne Jørgensen</i>	63
<i>Manzoor Ahmed Hashmani</i>	225
<i>Mathieu Febvay</i>	27
<i>Meghashyam Ashwathnarayan</i>	17
<i>Mohamed Mohanna</i>	341
<i>Moke Kwai Cheong</i>	225
<i>Mridula Prakash</i>	41, 73
<i>Nidal Faour</i>	01
<i>Ning Luo</i>	55, 91
<i>Ofer Shinaar</i>	01
<i>Pawan Nayak</i>	47
<i>Peter Schneider-Kamp</i>	253
<i>Ping An</i>	329
<i>Prasang Gupta</i>	197
<i>Priyanka Addagudi</i>	161

<i>Pu-Jen Cheng</i>	145
<i>Qinglin He</i>	329
<i>R.Manoj</i>	47
<i>Riccardo Occa</i>	207
<i>Richard Fulton</i>	127
<i>Robert Louka</i>	183
<i>Salem Alelyani</i>	341
<i>Sampath Shantanu</i>	47
<i>Shaojie Chen</i>	237
<i>Shaoxiang ZHOU</i>	191
<i>Srishti Mehra</i>	183
<i>Susan Kaplan</i>	127
<i>Vaishnavi J</i>	17
<i>Wendy MacCaull</i>	161
<i>Xiangyu XIA</i>	191
<i>Xiao Shan</i>	273
<i>Yahya Aseri</i>	341
<i>Yang Liu</i>	317
<i>Yixun Zhang</i>	183
<i>Yosi Ben-Asher</i>	01
<i>Yu Sun</i>	317
<i>Yuan Shi</i>	305
<i>Yue Xiong</i>	55
<i>Yuting Xue</i>	273
<i>Yuxuan Ding</i>	273