David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

11th International Conference on Computer Science and Information Technology (CCSIT 2021), May 29~30, 2021, Vancouver, Canada.

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: 978-1-925953-41-1
DOI: 10.5121/csit.2021.110701 - 10.5121/csit.2021.110718

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

Preface

The 11th International Conference on Computer Science and Information Technology (CCSIT 2021), May 29~30, 2021, Vancouver, Canada, 9th International Conference on Signal, Image Processing and Pattern Recognition (SIPP 2021), 10th International Conference on Parallel, Distributed Computing Technologies and Applications (PDCTA 2021), 9th International Conference on Artificial Intelligence, Soft Computing (AISC 2021), 2nd International Conference on Natural Language Processing & Computational Linguistics (NLPCL 2021), 2nd International conference on Big Data, Machine learning and Applications (BIGML 2021) and 6th International Conference on Networks, Communications, Wireless and Mobile Computing (NCWMC 2021) was collocated with 11th International Conference on Computer Science and Information Technology (CCSIT 2021). 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 CCSIT 2021, SIPP 2021, PDCTA 2021, AISC 2021, NLPCL 2021, BIGML 2021 and NCWMC 2021 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, CCSIT 2021, SIPP 2021, PDCTA 2021, AISC 2021, NLPCL 2021, BIGML 2021 and NCWMC 2021 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 CCSIT 2021, SIPP 2021, PDCTA 2021, AISC 2021, NLPCL 2021, BIGML 2021 and NCWMC 2021.

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

Organization

David C. Wyld, Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds) Wireilla Net Solutions, Australia

Program Committee Members

Abdel-Badeeh M. Salem, Ain Shams University, Egypt
Abdelhafid ZEROUAL, University of Artois, France
Abdelkaher Ait Abdelouahad, Chouaib Doukkali University, Morocco
Abdellatif I. Moustafa, Umm AL-Qura University, Saudi Arabia
Abderrahim Siam, University of Khenchela, Algeria
Abdulhamit Subasi, Effat University, Saudi Arabia
Adnan Albar, King AbdulAziz University, Saudi Arabia
Ahmad A. Saifan, Yarmouk University, Jordan
Ahmad Yarahmadi, Tarbiat Modares University, Iran
Akhil Gupta, Lovely Professional University, India
Alaa Hussein Al-hamami, Amman Arab University, Jordan
Alessio Ishizaka, NEOMA Business School, France
Alexander Gelbukh, Instituto Politécnico Nacional, Mexico
Ali Abdrhman Mohammed Ukasha, Sebha University, Libya
Ali Al-Sabbagh, Ministry of Communication, Iraq
Alok Jain, Samrat Ashok Technological Institute, India
Alyssa Gonzalez, University of Baghdad, Iraq
Amel Ourici Badji Mokhtar, University of Annaba, Algeria
Amer AbuAli, Taibah University, Saudi Arabia
Amin Karami, University of East London (UEL), London UK
Amina El murabet, Abdelmalek Essaadi University, Morocco
Amizah Malip, University of Malaya, Malaysia
Ana Leal, University of Macau, China
Anand Nayyar, Duy Tan University, Vietnam
Andrej V. Plotnikov, Odessa State Academy, Ukraine
Anita Patil, Cummins college of Engineering for Women, India
Ann Zeki Ablahd, Northern Technical University, Iraq
Archana Mire, Terna Engineering College, India
Archit Yajnik, Sikkim Manipal University, India
Arjav Bavarva, RK University, India
Ashraf Elnagar, College of Computing and Informatics, UAE
Assem Mousa, E Commerece Tech Support Systems Manager, Egypt
Assia DJENOUHAT, Universty Badji Mokhtar Annaba, Algeria
Attila Kertesz, University of Szeged, Hungary
Ayush Singhal, Contata Solutions, USA
Azeddine Wahbi, Hassan II University, Morocco
Baihua Li, Loughborough University, UK
Beshair Alsiddiq, Prince Sultan University, Saudi Arabia
Bin Zhao, Xidian University, China
Bo-Cheng La, National Chiao Tung University, Taiwan
Bouchra Marzak, Hassan II University, Morocco

boukari nassim,	skikda university, algeria
BRAHAMI Menaouer,	Qatar University, Qatar
Casalino Gabriella,	University of Bari, Italy
Christian Mancas,	Ovidius University, Romania
Christos Bouras,	University of Patras, Greece
Claude Tadonki,	MINES ParisTech, France
Claudio Gallicchio,	University of Pisa, Italy
Dário Ferreira,	University of Beira Interior, Portugal
Diab Abuaiadah,	Waikato Institute of Technology, New Zealand
Divya Saxena,	IIT Roorkee, India
Domenico Rotondi,	FINCONS SpA, Italy
Dongping Tian,	Baoji University of Arts and Sciences, China
Edwin Lughofer,	Johannes Kepler University Linz, Austria
Elżbieta Macioszek,	Silesian University of Technology, Poland
Eng Islam Atef,	Alexandria University, Egypt
Eng Islam Atef,	Engineering alexandria uinversity, Egypt
Farzin Piltan,	University of Ulsan, Korea
Felix J. Garcia Clemente,	University of Murcia, Spain
Fernando Zacarias Flores,	Universidad Autonoma de Puebla, Mexico
Fulvia Pennoni,	University of Milano-Bicocca, Italy
G. A. Walikar,	Walchand Institute of Technology, India
Giuliani Donatella,	University of Bologna, Italy
Giuseppe Di Modica,	University of Catania, Italy
Grigorios N. Beligiannis,	University of Patras, Greece
Grzegorz Sierpiński,	Silesian University of Technology, Poland
Hala Abukhalaf,	Palestine Polytechnic University, Palestine
Hamid Khemissa,	USTHB University Algiers, Algeria
Hamid Mcheick,	Université du Québec à Chicoutimi, Canada
Hamidah Ibrahim,	Universiti Putra Malaysia, Malaysia
Hanan Salam,	University of Pierre and Marie Curie, France
Hao-En Chueh,	Chung Yuan Christian University, Taiwan
Hao-En Chueh,	Yuanpei University, R.O.C
Harto Tanujaya,	Tarumanagara University, Indonesia
Hashem Ramadan,	Indian academy Degree College, India
Hassan,	II University, Morocco
Heba Elgazzar,	Morehead State University, USA
Himani mittal,	GGDSD College, India
Hiromi Ban,	Nagaoka University of Technology, Japan
Homero Toral Cruz,	University of Quintana Roo, México
Ihab Zaqout,	Al-Azhar University - Gaza, Palestine
Ilham Huseyinov,	Istanbul Aydin University, Turkey
Ines Bayoudh Saadi,	Tunis University, Tunisia
Ioannis Karamitsos,	University of Aegean, Greece
Irina Perfilieva,	University of Ostrava, Czech Republic
Israa Shaker Tawfic,	Ministry of Science and Technology, Iraq
Iyad Alazzam,	Yarmouk University, Jordan
J.Naren,	SASTRA Deemed University, India
Jagadeesh HS,	APS College Of Engineering, India
Janusz Kacprzyk,	Systems Research Institute, Poland
Javid Taheri,	Karlstad University, Sweden
Jesuk Ko,	Universidad Mayor de San Andres (UMSA), Bolivia
Jia Ying Ou,	York University, Canada

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

11th International Conference on Computer Science and Information Technology (CCSIT 2021)

9th International Conference on Signal, Image Processing and Pattern Recognition (SIPP 2021)

10th International Conference on Parallel, Distributed Computing Technologies and Applications (PDCTA 2021)

9th International Conference on Artificial Intelligence, Soft Computing (AISC 2021)

2nd International Conference on Natural Language Processing & Computational Linguistics (NLPCL 2021)

2nd International conference on Big Data, Machine learning and Applications (BIGML 2021)

6th International Conference on Networks, Communications, Wireless and Mobile Computing (NCWMC 2021)

FACEATLASAR: ATLAS OF FACIAL ACUPUNCTURE POINTS IN AUGMENTED REALITY

Menghe Zhang[1], Jürgen P. Schulze[1] and Dong Zhang[2]

[1]Department of Computer Science, University of California, San Diego, USA
[2]Qilu University of Technology, Shandong, China

ABSTRACT

Acupuncture is a technique in which practitioners stimulate specific points on the body. Those points, called acupuncture points (or acupoints), anatomically define areas on the skin relative to specific landmarks on the body. However, mapping the acupoints to individuals could be challenging for inexperienced acupuncturists. In this project, we proposed a system to localize and visualize facial acupoints for individuals in an augmented reality (AR) context. This system combines a face alignment model and a hair segmentation model to provide dense reference points for acupoints localization in real-time (60FPS). The localization process takes the proportional bone (B-cun or skeletal) measurement method, which is commonly operated by specialists; however, in the real practice, operators sometimes find it inaccurate due to the skill-related error. With this system, users, even without any skills, can locate the facial acupoints as a part of the self-training or self-treatment process.

KEYWORDS

Augmented reality, Acupuncture point, Face alignment, Hair segmentation.

1. INTRODUCTION

Acupuncture [1] is a form of alternative medicine and a key component of traditional Chinese medicine (TCM). Based on the symptoms, acupuncturists stimulate specific anatomic sites commonly by needling, massaging, or heat therapy. Scientific studies have proved that acupuncture may help ease types of pain that are often chronic such as low-back pain, neck pain, and osteoarthritis/knee pain. The acupoints on the face can help with a variety of conditions both on and off the face, such as jaw tension, headaches, anxiety, and stomach conditions.

However, acupuncture practice relies on experienced acupuncturists to locate the acupoints from body acupuncture maps. Individuals, who want to help themselves relieve symptoms with acupoints stimulation, usually find it confusing to localize the targets by natural language description or pictures of a standard model.

With the help of augmented reality, we designed a system to display facial acupoints on the top of the user's face to accurately view and localize facial acupuncture points. Specifically, we employ a deep learning model to annotate 3D landmarks and on a user's face together with hair segmentation in real-time to gather reference points for acupuncture points localization. We then align the reference point with acupuncture points defined by proportional bone measurement method. The whole process is implemented via MediaPipe [2], a framework for building cross-platform machine learning solutions. Our system works perfectly on phones with a front camera,

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 01-11, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110701

without any extra hardware, users could pinpoint and interact with the target acupoints. There are three benefits to this application:

- It takes a conventional localization method while originally defines a scheme to transform the natural language descriptions to mathematical logic expressions.
- It adopts MediaPipe framework to run across platforms in real-time.
- It adapts to different head poses to be robust for users to locate acupoints in different regions.

With FaceAtlasAR, people who have little or no experience in localizing facial acupuncture points can use. Potential use scenarios for our applications are varied, for example, acupuncture education, communication, and self-healing.

2. RELATED WORK

2.1. Acupuncture Points Localization

Existing works of acupuncture training applications on AR devices are limited. In 2015, H. Jiang et al. [3] proposed the first acupuncture training application, Acu Glass, on a head-mount display device (HMD) based on Google Glass. They generated the frontal face acupoints based on the height and the width of the input face, plus the distance between the eyes. However, their face landmarks for reference are too limited to adapt to different people and different poses of the face. Other acupoints localization methods like Chen et al. [4][5] fit a 3D Morphable Face Model(3DMM) [6] to a 2D image and combine facial landmarks and image deformation to estimate acupoints. Although 3DMM is a powerful tool to build polygonal mesh, the range of possible predictions is limited by the linear manifold spanned by the PCA basis, which is in turn determined by the diversity of the set of faces captured for the model [7]. Therefore, manual annotation on a standard 3D model may not correctly fit all kinds of people. Moreover, acupoints are officially defined relative to landmarks, while the deformation process does not guarantee the relativeness.

As for the localization methods in practice, Godson and Wardle [8] screened 771 studies and summarized the methods as Directional(F-cun) method, Proportional method, palpation for tenderness, electronic point detectors, and anatomical locations. Usually, more accurate approaches are the next steps to less accurate ones and require extra hardware. For example, one can roughly locate a target by the directional method and then use electronic point detectors to measure the electrical resistance of the skin. There is research related to this topic and acupoint probing devices available in the market.

2.2. Face Alignment

Face alignment is a computer vision technology for identifying the geometric structure of human faces in digital images. Bulat and Tzimiropoulos [9] reviewed 2D and 3D face alignment and landmark localization. Existing 2D and 3D datasets annotate a limited set of landmarks. For example, 300-W [10], the most widely used in-the-wild dataset for 2D alignment, containing LFPW [11], HELEN [12], AFW [13], and iBUG [10], annotates only 68 landmarks per face. These landmarks either have distinct semantics of their own or participate in meaningful facial contours. Works based on them are not suitable for our cases. We finally adopted the work of Kartynnik et al. [7], which estimates 3D mesh with 468 vertices in real-time. The vertices are selected manually according to expressive AR effects, thus well suitable for our requirements.

3. SYSTEM OVERVIEW

Our facial localization solution utilizes the MediaPipe machine learning pipeline consisting of an off-line stage and an on-line stage (Fig.1).

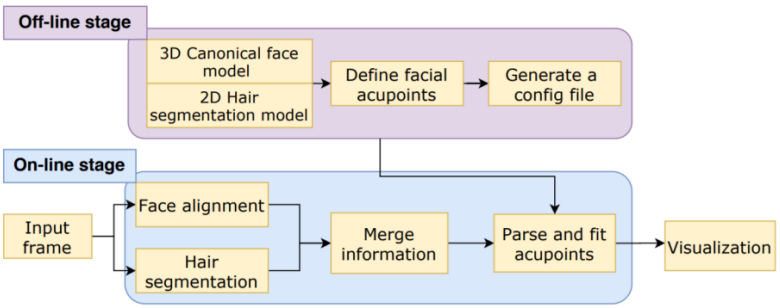


Figure 1: Workflow of the proposed system

During the off-line stage:

- **Model selection**: Based on the definition of the referenced points on a face, we choose to blend a pre-trained face alignment model and a pre-trained hair segmentation model. This is because a single model cannot cover the whole set of target regions, while different parts of the regions are in the different reference systems.
- **Acupoints localization**: We firstly localize facial anatomical landmarks, which are designated by the National Standard of the People's Republic of China. We then use the proportional bone (B-cun or skeletal) measurement method to locate all acupoints on the face.
- **Data file generation**: The file contains all the information needed for each acupoint/reference point, including name, region, relative location towards a landmark or a reference point. The file is readable for non-technical users, for example, acupuncturists, to correct less accurate acupoint descriptions by natural language.

Then at the on-line stage, the system gets face landmarks together with hair segmentation and merges those results into acupoints generator. The generator gives out the requested acupoints on this frame based on the prior knowledge and then draw on the input face.

We fit the whole process into MediaPipe's perception pipeline as a graph of modular components. Each component, called Calculator, solves an individual task like model inference, data transformation, or annotation. We will talk about the implementation details in the next section. We show the graph for our FaceAtlasAR in Figure 2.

The graph consists of two subgraphs: one for face alignment (FaceLandmarkFrontGpu) and the other for hair segmentation (HairSegmentationGpu). From the graph, we see the FlowLimiter guards the whole pipeline process. By connecting the output of the final image to the FlowLimiterwith a backward edge, the FlowLimiter keeps track of how many timestamps are currently being processed. The system will down-sample and transform the original image before fusing the input to machine learning models but will draw the results onto the original frames. The next section will show the implementation of each module in detail.

4. IMPLEMENTATION

4.1. Face Alignment

We adopted a pre-trained TFLite model [7] to infer an approximate 3D mesh representation of a human face.

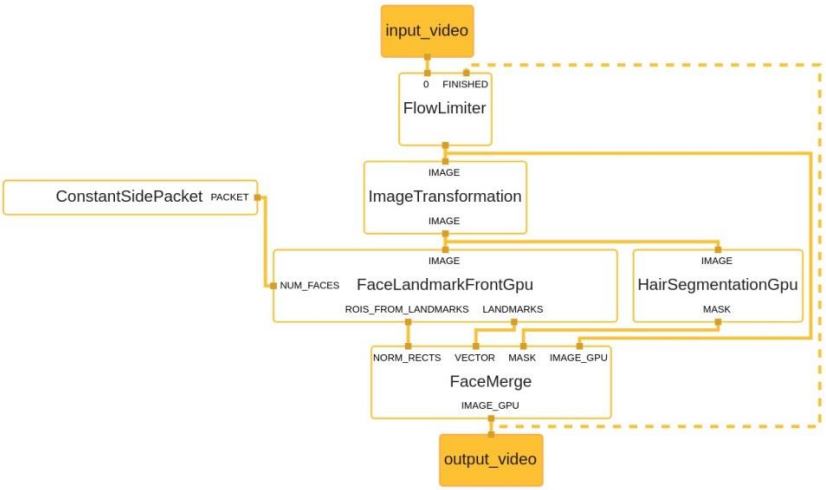


Figure 2: The graph of FaceAtlasAR

This process comprises of majorly three steps:

- **Face detection**: The whole frame is first processed by a lightweight face detector to get the face bounding box and several landmarks, thus, to get the rotation matrix of the face. This step only runs until the system finds a face to track or when the system loses tracking.
- **Image transformation**: The image is then cropped by the bounding box and resized to fit into the next step. After this step, the target region is centered and aligned.
- **Face landmarks generation**: The pre-trained model produces a vector of 3D landmark coordinates, which subsequently gets mapped back into the original image coordinate system.

Then from a canonical face mesh model, we extract those vertices with semantic meaning as the reference points (Fig.3).

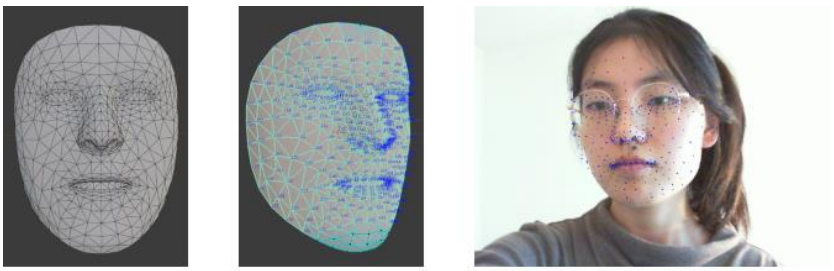


Figure 3: The generated mesh topology(a), its vertices with index(b), and viewed in AR(c)

4.2. Hair Segmentation

Since the center of the frontal hairline is a critical facial anatomical landmark according to the national standard, we adopted a pre-trained model [14] to get the hair segmentation mask. Similar to the face alignment process, the previously generated mask can be fed back to help accelerate the process. Specifically, the mask from the previous round of inference will be embedded as the alpha channel of the current input image (Fig.4).

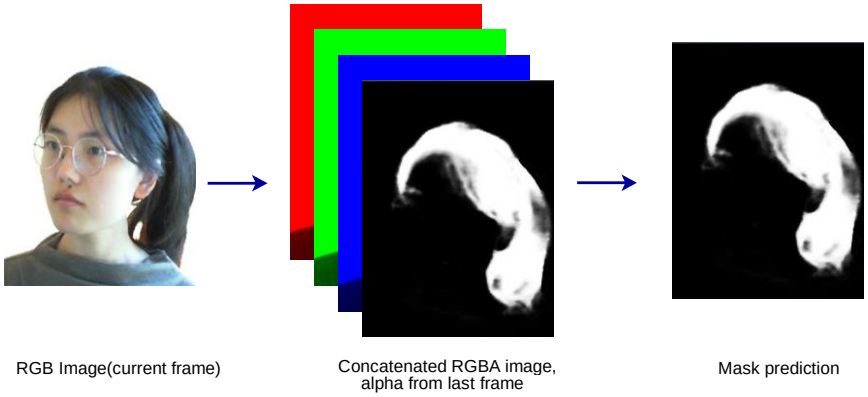
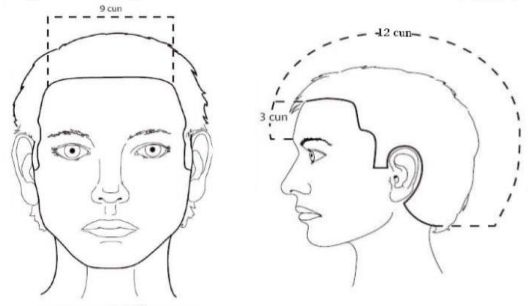


Figure 4: Hair segmentation module

4.3. Facial Acupoints Localization

Given the hair mask together with face landmarks, we now could locate facial acupoints based on the B-cun method. This refers to the method of measuring the length and width of each part of the body with the body surface condyles as the main landmark and determining the position of acupoints. Then, a unit "cun" is the length between the set two bone nodes divided into certain equal parts as the basis for setting acupoints. The facial acupoints bank on the unit cun's definition of the head as shown in Figure 5. To start with, we locate three facial anatomical landmarks in consonance with the national standard. In order to differentiate these three points to acupoints, we group them in the channel named RHD:

- **RHD1, Yintang:** The midway between the medial ends of the eyebrows.
- **RHD2, Middle of anterior hairline:** Intersection of anterior hairline and anterior midline.
- **RHD3, Pupils**



(a) B-cun on the font face (b) B-cun on the side face

Figure 5: B-cun on the head from National Standard of the People's Republic of China, Acupoints [15]. Pictures from [16].

Table 1: Information of an example acupoint Sibai (ST2) in the data file.

Channel Name	ID	NameE	Region
ST	2	Sibai	eye
FaceMeshX	FaceMeshY	IsSymmetry	Comments
GetX(RHD3)	GetY(ST1)+0.5*U	TRUE	-

From Fig.5b, the distance from Yintang to Middle of anterior hairline, d= (p_{RHD1} -p_{RHD2}), decides the unit cun as $uc = d/3$.

On that occasion, like RHD points definition, we could locate all facial acupoints. Table1 shows an example of what information we keep for each acupoint. All points' information finally makes up to a data file.

There is one more aspect we want to specify, the channel name, which refers to a unique meridian channel. The meridian system[17] is a concept in TCM about a path through which the life-energy is known as "qi" flows. There are 12 standard and 8 extraordinary meridians, while acupoints arethe chosen sites on the meridian system. We group the acupoints on the same meridian channel and connect them by the flow. For example, the previously stated acupoint, Sibai, belongs to the ST(Foot's Yang Supreme Stomach Meridian) channel. Acupoints on the ST channel and their flow are illustrated in the Figure 6.

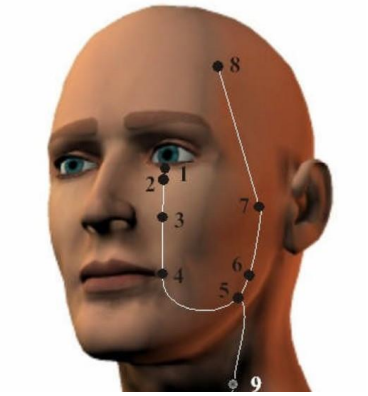


Figure 6: Illustration of ST channel on the head. Picture from [18]

5. RESULTS

We achieve real-time performance on both desktop and mobile devices by designing the pipeline properly. We show the final application below and compare the performance on different platforms.

5.1. Android Application

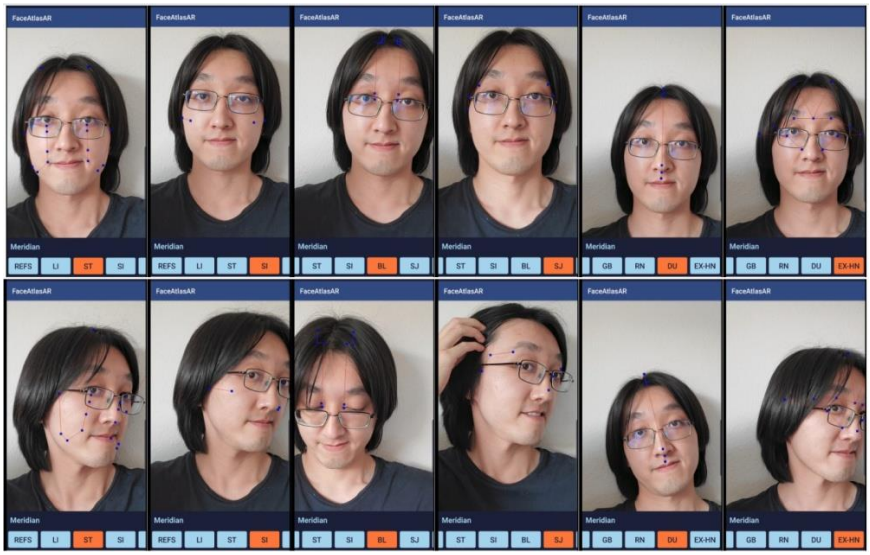


Figure 7: Android application screenshots for displaying acupoints grouped by meridian system in different poses

Figure 7 presents the screenshots from our FaceAtlasAR android app. Here we show the visualization of acupoints grouped by meridian channels in different poses. Thanks to the robustness of face alignment towards occlusion, users would not find problems pointing or pressing a target acupoint (Fig. 8).

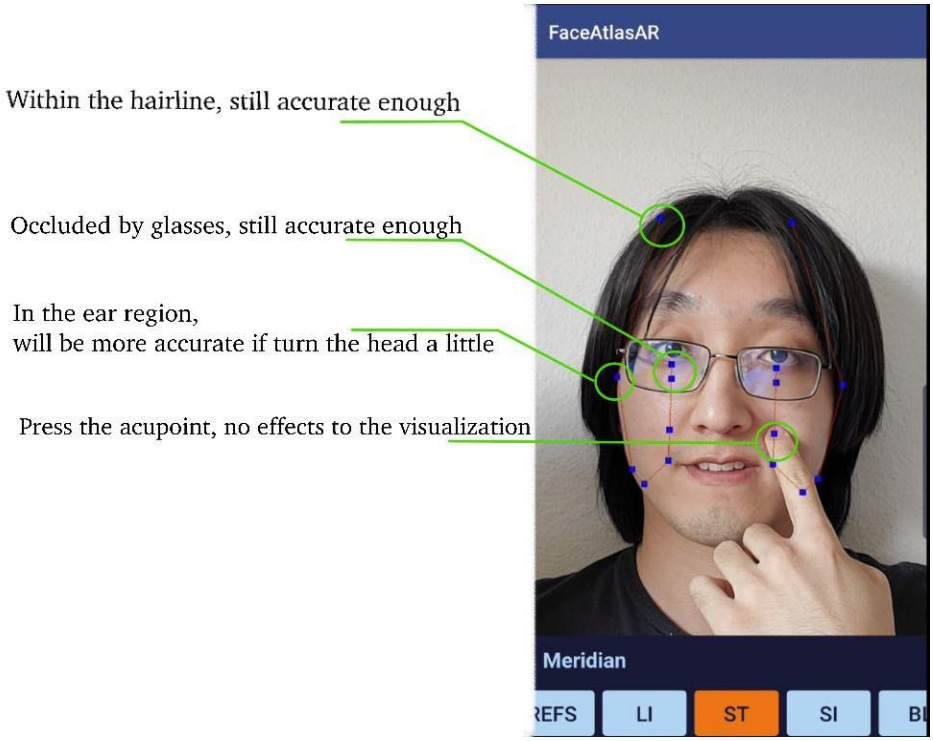


Figure 8: User presses an acupoint on ST meridian channel

5.2.Accuracy

We setup an experiment on Samsung S10(2280 × 1080) to valid the accuracy of our FaceAtlasAR to localize 4 reference points and 69 acupuncture points on face by comparing the mean pixel errors between ground truth positions and the localization results. Initially, we assign all the target positions into three groups by the localization complexity as shown in the Table 2.

Table 2: Data file parsing performance

Quantity	Directly from face alignment results	One-time proportional localization	Multiple-times proportional localization
Reference points	3	1	0
Acupoints	38	16	15

We then investigate the mean pixel errors in 3 groups. For each localization points, we measure the pixel errors in 4 different poses: frontal face (0^o), pitch (X-axis $+10^o$), roll (Y-axis $+10^o$), yaw (Z-axis $+10^o$) to get the mean value. The results are shown in Figure 9.

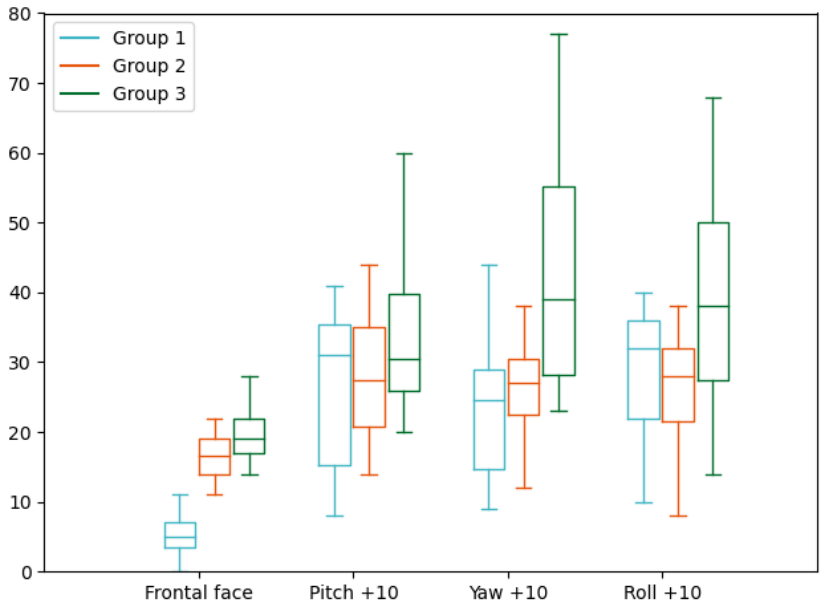


Figure 9: Mean pixel errors of localization

From the results we found that some of the localizations are less precise than others. Especially, the system performs worse on group 2 than group 1 and group 3 than group 2. Accordingly, multiple times of proportional calculations that rely on the cun system add inaccuracy to the results. The system tolerates to different angles to some extent. However, a target point will be more well localized when it is point straight to the camera without any angles.

5.3. Performance

We evaluate the performance of our pipeline in three major components: face alignment, hair segmentation, and acupoints generation. Since the application runs on multiple platforms, we compare its performance on a desktop with Nvidia GeForce RTX 2070 SUPER and on a

Samsung S10. The input images for two TFLite models are both in full size 512×512. We see that in this case, Samsung S10 still runs in full frame rate (60FPS). More detailed comparison is as in Table 3. We also evaluate the performance of data file parsing(Table 2), which only runs once at the setup stage.

Table 3: Data file parsing performance

Device/Time, ms	File parser
Desktop	0.102
Samsung S10	0.279

Table 4: Application performance on a desktop and a mobile device

Device/Time(ms)	Hair segmentation	Face alignment	Generation	Overall
Desktop	1.188	3.376	0.135	10.56
Samsung S10	50.279	14.673	0.512	84.758

6. DISCUSSION

From the results we could see that:

- Our system can properly display the requested acupoints on selected meridian channels.
- The system tolerates movements very well, while endures tilt and rotation to some degree.
- The benefit from head rotation is that: when the acupoints are hidden in one view, they will be visible and more accurate in another one. For example, the frontal face hides acupoints in the ear region; thus, to view them around the left ear properly, the user needs to turn his/her head so that the left ear faces the camera.

Our next step is to improve the face alignment performance on the side of the head. There are dozens of acupuncture points around the ears that represent specific domain and functions of the body. However, most face alignment jobs only require a small set of landmarks. Even though the model we adopted can estimate 3D mesh with 468 vertices, it still neglects both sides of the head and loses some accuracy at the cheeks and chins. Therefore, we could only roughly estimate those acupoints' positions based on the proportional relations to other landmarks on face, which is less meticulous.

7. CONCLUSION

In this paper, we proposed FaceAtlasAR, an end-to-end facial acupoints tracking solution that achieves real-time performance on mobile devices. Our pipeline integrates a face alignment model with a hair segmentation model. The high accuracy of the estimation and the robustness of the system empower users with little experience in acupuncture to interact with facial acupoints. Future work comes to improving the accuracy even further in the ear region since the face alignment only gives a little information towards the face edge near the ear region. Also, 3D interaction should be considered for users to gain a more immersive experience. For example, we could track users' hands/fingers while they are interacting with a target acupoint.

REFERENCES

[1] Wikipedia. *Acupuncture*. https://en.wikipedia.org/wiki/Acupuncture#cite_note-3. 2020.

[2] Camillo Lugaresi et al. "Mediapipe: A framework for building perception pipelines". In: *arXiv preprint arXiv:1906.08172* (2019).

[3] Fang Li et al. "What is the Acupoint? A preliminary review of Acupoints". In: *Pain Medicine* 16.10 (2015), pp. 1905–1915.

[4] Yi-Zhang Chen et al. "Localization of Acupoints using Augmented Reality". In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. 2017, pp. 239–241.

[5] Kun-Chan Lan, Guan-Sheng Li, and Jun-Xiang Zhang. "Toward Automated Acupressure Therapy". In: *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. 2018, pp. 384–385.

[6] Patrik Huber et al. "A multiresolution 3d morphable face model and fitting framework". In: *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. 2016.

[7] YuryKartynnik et al. "Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs". In: *arXiv preprint arXiv:1907.06724* (2019).

[8] Debra R Godson and Jonathan L Wardle. "Accuracy and precision in acupuncture point location: a critical systematic review". In: *Journal of acupuncture and meridian studies* 12.2 (2019), pp. 52–66.

[9] Adrian Bulat and Georgios Tzimiropoulos. "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1021–1030.

[10] Christos Sagonas et al. "A semi-automatic methodology for facial landmark annotation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2013, pp. 896–903.

[11] Peter N Belhumeur et al. "Localizing parts of faces using a consensus of exemplars". In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2013), pp. 2930– 2940.

[12] Vuong Le et al. "Interactive facial feature localization". In: *European conference on computer vision*. Springer. 2012, pp. 679–692.

[13] Xiangxin Zhu and Deva Ramanan. "Face detection, pose estimation, and landmark localization in the wild". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 2879–2886.

[14] Andrei Tkachenka et al. "Real-time Hair Segmentation and Recoloring on Mobile GPUs". In: *arXiv preprint arXiv:1907.06740* (2019).

[15] Unknown. *National Standard of the People's Republic of China, Acupoints*. https:// musculoskeletalkey.com/ynsa-basic-points/. 2005.

[16] Unknown. *YNSA Basic Points*. https://musculoskeletalkey.com/ynsa-basic-points/. 2017.

[17] Wikipedia. *Meridian (Chinese medicine)*. https://en.wikipedia.org/wiki/Meridian_ (Chinese_medicine). 2020.

[18] Unknown. *Atlas of Acupuncture Points*. www.AcupunctureProducts.com. 2007.

AUTHORS

Menghe Zhang
Graduate student, University of California, San Diego

Jürgen P. Schulze
Associate Research Scientist, Qualcomm Institute, UCSD
Associate Adjunct Professor, Department of Computer Science, UCSD

Dong Zhang
Research Scientist, Qilu University of Technology

THREAT ACTION EXTRACTION USING INFORMATION RETRIEVAL

Chia-Mei Chen[1], Jing-Yun Kan[1], Ya-Hui Ou[2],
Zheng-Xun Cai[1] and Albert Guan[3]

[1]Department of Information Management,
National Sun Yat-sen University, Taiwan
[2]National Penghu University of Science and Technology, Taiwan
[3]Department of Applied Mathematics, National Sun Yat-sen University, Taiwan

ABSTRACT

To gain insight into potential cyber threats, this research proposes a novel automatic threat action retrieval system, which collects and analyzes various data sources including security news, incident analysis reports, and darknet hacker forums and develops an improved data preprocessing method to reduce feature dimension and a novel query match algorithm to capture effective threat actions automatically without manually predefined ontology applied by the past research. The experimental results illustrate that The proposed method achieves an accuracy of 94.7% and a recall rate of 95.8% and outperforms the previous research. The proposed solution can extract effective threat actions automatically and efficiently.

KEYWORDS

cyber threat intelligence, word vector, information retrieval.

1. INTRODUCTION

Organizations and businesses apply modern information technologies to expand services and improve customer satisfaction, while in the meantime they are facing potential cyberattacks. Cyberattacks have increased in frequency and sophistication, presenting significant challenges for organizations that must defend their data and systems from capable threat attackers. They utilize a variety of tactics, techniques, and procedures (TTPs) to compromise systems, disrupt services, commit financial fraud, and expose or steal intellectual property and other sensitive information. Given the risks these threats present, organizations seek solutions to improve information security and reduce cyberattack risks.

TTPs are the patterns of activities or methods associated with a specific threat actor or group of threat actors [1], which help to identify common attack vectors and possible vulnerable systems likely compromised. Among the key elements of TTP information, identify threat actions is the most essential for understanding TTPs and proactively defending against cyberattacks.

Machine learning techniques have been applied to CTI research recently. Most past research focused on classifying security and non-security related documents or extracting vulnerabilities [2-5] but rarely extracting attack tactics to fill up the information needed by APT incidents to outline attack processes. Some previous work [6-8] manually built up a TTP ontology that consumes intensive labor work and requires to keep it updated as new attack vectors emerge.

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 13-19, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110702

To obtain efficient threat actions, such as hide malicious operations, avoid raising suspicion, and contain .scr file, cybersecurity staff needs to acquire a wide range of articles in order to comprehend the information. Based on the reading speed statistics from ExecuRead [9], the reading speed of technical articles is 50~75 wpm, which takes 5 ~ 6 minutes per page. For a mid-size APT report [10] of 12 pages, a reader needs one hour or so to comprehend the threat actions in the report and may miss some. Such a task is labor-intensive and desires an efficient and automatic threat action retrieval method.

To our best knowledge, the present study is the first attempt to automatically identify threat actions without manually defined ontology by applying multiple word vector models. This research proposes a CTI retrieval method that extracts a key threat action list, which replaces the role of ontology applied by the previous research. Furthermore, the proposed method develops a new query match algorithm that combines multiple word vector language models and similarity functions to capture effective threat actions automatically.

The primary contribution of this study is discovering potential cybersecurity information by exploring multiple types of data sources and multiple state-of-the-art word vector models and developing a novel information retrieval method that extracts threat actions automatically without ontology.

The remainder of the paper is structured as follows. Section 2 reviews the state of the art in the scope of threat intelligence extraction and natural language processing approaches. Section 3 presents the proposed threat action extraction method, followed by the performance evaluation and discussion in Section 4. The last section draws the conclusion remark and the future directions of this study.

2. LITERATURE REVIEW

Settanni et al. [11] evaluated their proposed document correlation methods, where a document is represented as a feature vector, and demonstrated that features based on TF-IDF from the document's own words perform better and those from pre-determined dictionary exhibit a low accuracy and precision.

Niakanlahiji et al. [12] employed a context-free grammar (CFG) model to extract candidate threat actions and applied TF-IDF to extract threat actions. Their results imply that TF-IDF is suitable for representing the importance of a candidate threat action among a list of tokens, so this study adopts it for extracting relevant short phrases from candidate threat actions.

Distributed representations of words in a vector space help learning algorithms to achieve better performance in NLP tasks by grouping similar words. Word2Vec (W2V) [13] is a family of word embedding (word vector) models of representing distributed representations of words in a corpus, where Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model are commonly used. Word2Vec is a two-layer neural network and produces a vector space, where each unique word in a corpus is assigned a corresponding vector in the space.

A study [14] concluded that Word2Vec outperforms the traditional feature selection models including CHI, IG, and DF. As words may have different meanings (i.e., senses) depending on the context, identifying words in the correct meaning is important for extracting relevant information. Two previous studies [15, 16] concluded that Cosine similarity and Word2Vec can effectively capture syntactic word similarities and outperforms LSA (Latent semantic analysis) commonly used in word sense disambiguation. Both applied WordNet [17] as the evaluation

corpus. WordNet is a large lexical database of English, where nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Word2Vec models lose the ordering of the words and ignore the semantics of the words. An unsupervised algorithm Doc2Vec (D2V) [18] represents each document by a dense vector, which overcomes the weaknesses of Word2Vec. Kadoguchi et al. [19] applied Doc2Vec and ML technology to classify information security data from dark web forums, and the results indicate that Doc2Vec is effective on feature selection and a multi-layer classifier can achieve 79% accuracy. Another study [20] applied Doc2Vec with Cosine similarity on classifying court cases and yields 80% accuracy. A performance study [20] demonstrated that Word2Vec and Doc2Vec perform better than N-gram on text classification and semantic similarity.

If a word is not in the training corpus, Word2Vec fails to identify its similar words. FastText [21, 22] improves the drawback of Word2Vec by applying N-gram to build on not just using the words in the training vocabulary but also their substrings. FastText became popular and replaced Word2Vec on text classification [23, 24] after it was invented. A study demonstrated that FastText achieves 78% accuracy better than Word2Vec and Doc2Vec on text classification; another study [25] drew a similar conclusion remark.

TTPDrill [6] adopted Stanford typed dependency parser to extract candidate threat actions and then mapped these candidate threat actions to those in a pre-defined ontology based on BM25 [26] similarity score. A follow-up study, ActionMiner, [8] improved the above parser of candidate action extraction by applying entropy and mutual information to understand the specificity of verbs used in cybersecurity reports. A study [7] manually selected threat actions, classified the features of the selected threat actions, and associated the threat actions and malware by random forest. The above research all depend on an ontology defined manually and labor-intensive. This study proposes an automatic method to construct a key threat action list in replacement of a manually defined ontology and employs ML technology to analyze and identify effective threat actions.

3. THE PROPOSED METHOD

Figure 1 overviews the major components of the proposed CTI retrieval method. In the model building, it retrieves documents from the security-related websites and sanitizes the text content, and then labels all the tokens by applying the part-of-speech tagging method, extracts verb-form tokens as a candidate threat action list.

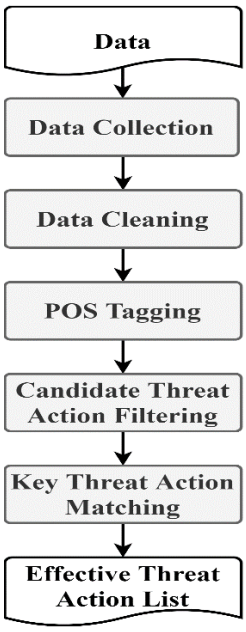


Figure 1. The architecture of the proposed method (source from this study).

Common NLP techniques are adopted to reduce the feature dimension, including tokenization, stop word removal, stemming, and lemmatization. Data cleaning reduces the word density in a given text and helps in preparing the accurate features for model training, as cleaned data improves the efficiency of ML models. The candidate threat actions are extracted from the above-cleaned text by applying POS tagging.

As an ontology requires intensive labor work and its efficiency heavily relies on the completeness of the human-defined ontology, the proposed method plans to automatically build up a key threat action list that is the key component of a TTP ontology to reduce the dependency on domain knowledge. To construct a key threat action list, a two-stage process is developed: the first stage filters out non-security related action tokens and the second stage applies similarity matching measure to extract key threat actions, where the key threat action list serves the purpose of the ontology used in the past research in threat action retrieval.

4. SYSTEM VALIDATION AND EVALUATION

The dataset is sourced from information security reports, Github's APTNotes [27], where APTNotes have acquired comprehensive APT attack investigation reports published by cybersecurity companies, which explain attack chains and threat actions in detail. A total of 600 articles published from 2008 to May 2020 has been collected, where 520 reports from 2008 to 2019 are used for training and the newer ones are for evaluation purpose in order to evaluate if the proposed system can identify threat actions effectively based on a past dataset.

The articles in the dataset are studied, and the threat actions of each article are labeled by a security professional for performance evaluation, where the labeled dataset is summarized in Table 1.

Table 1. The threat actions of the dataset (source from this study).

Dataset	No. of threat actions			No. of articles for testing/training
	Candidate	Effective	Invalid	
APTNotes	21,631	1,438	20,193	95/520

To validate the proposed method by comparing compare the performance of different combinations of filtering and similarity methods as listed in Table 2. The experimental results demonstrate that the proposed threat action retrievalmethod achieves the best performance among all the different combinations of filtering and similarity methods and can identify threat actions effectively.

Based on our preliminary study, a security-related article might contain non-security action words that are not related to threat actions. An ontology-based approach requires intensive manual work. Therefore, the study proposes a multi-stage threat action retrieval approach in order to minic the effort of ontology. The data clean removes noise and consolidates synonyms; the POS tagging labels and filters out non-verb tokens; the filtering process removes non-security action verbs; the matching process computes the similarity of the threat actions to retrieve key threat actions. The correctness of the experimental results are validated by humans, and the results verify the study objective: automatically retrieving threat actions without ontology.

Table 2. The performance of the different filtering and matching methods (source from this study).

Filtering	Matching	Precision	Recall	F1-score
TF-IDF	BM25	71.63%	28.55%	40.83%
WordNet	BM25	62.78%	74.80%	68.00%
WordNet	BM25, W2V, D2V, FastText	90.63%	94.55%	92.58%

5. CONCLUSION

This study applies word vector, tagging, filtering techniques to capture threat actions. The novelty of the proposed solution includes automatically producing a key threat action list as the base of the ontology, the two-stage key threat action extraction algorithm, and applying word vector models for key threat extraction.

The experimental results demonstrate that the proposed solution can capture effective threat actions efficiently with high accuracy and outperforms the previous research. According to the results, the proposed method achieves the following research goals: to identify threat actions efficiently without a predefined ontology and to be able to extract threat actions from different types of documents and in different languages.

This study applies part-of-speech tagging to label tokens in a sentence with their grammatical word categories, but it does not maintain grammatical relations between them. The future work might be able to explore dependency parsing to analyze the sentence structure as it keeps tokens' grammatical relations.

As this research focuses on retrieving threat actions, other pieces of CTI information might be useful for attack prevention. Exploring the relationships among adversaries, victims, and threat actions is another possible research direction for understanding the correlations of these parties.

REFERENCES

[1] J. Friedman and M. Bouchard. "Definitive Guide to Cyber Threat Intelligence." https://cryptome.org/2015/09/cti-guide.pdf (accessed: Nov. 11, 2020).

[2] C. Sabottke, O. Suciu, and T. Dumitraș, "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits," in 24th Security Symposium (Security 15), 2015, pp. 1041-1056.

[3] A. S. Gautam, Y. Gahlot, and P. Kamat, "Hacker Forum Exploit and Classification for Proactive Cyber Threat Intelligence," in International Conference on Inventive Computation Technologies, 2019: Springer, pp. 279-285.

[4] L.-J. Wei, "Distinguishing between Intelligence Articles and Technical Articles based on Extracted Keywords and IOC Elements," Master, Dept. of Computer Science, National Taiwan University of Science and Technology, 2017.

[5] N. Dionísio, F. Alves, P. M. Ferreira, and A. Bessani, "Cyberthreat detection from twitter using deep neural networks," in 2019 International Joint Conference on Neural Networks (IJCNN), 2019: IEEE, pp. 1-8.

[6] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources," presented at the Proceedings of the 33rd Annual Computer Security Applications Conference, 2017.

[7] Z. Zhu and T. Dumitraș, "FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature," presented at the Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[8] G. Husari, X. Niu, B. Chu, and E. Al-Shaer, "Using entropy and mutual information to extract threat actions from cyber threat intelligence," in 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), 2018: IEEE, pp. 1-6.

[9] ExecuRead. "Speed Reading Facts." https://secure.execuread.com/facts/ (accessed: July. 3, 2020).

[10] YOROI. "The North Korean Kimsuky APT keeps threatening South Korea evolving its TTPs " https://yoroi.company/research/the-north-korean-kimsuky-apt-keeps-threatening-south-korea-evolving-its-ttps/ (accessed: Aug. 3, 2020).

[11] G. Settanni, Y. Shovgenya, F. Skopik, R. Graf, M. Wurzenberger, and R. Fiedler, "Acquiring cyber threat intelligence through security information correlation," in 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), 2017: IEEE, pp. 1-7.

[12] S. Chandel, J. Wei, and B.-T. Chu, "A natural language processing based trend analysis of advanced persistent threat techniques," in 2018 IEEE International Conference on Big Data (Big Data), 2018: IEEE, pp. 2995-3000.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, 2013, pp. 3111-3119.

[14] W. Tian, J. Li, and H. Li, "A method of feature selection based on Word2Vec in text categorization," in 2018 37th Chinese Control Conference (CCC), 2018: IEEE, pp. 9452-9455.

[15] K. Orkphol and W. J. F. I. Yang, "Word sense disambiguation using cosine similarity collaborates with Word2vec and WordNet," vol. 11, no. 5, p. 114, 2019.

[16] A. Handler, "An empirical study of semantic similarity in WordNet and Word2Vec," 2014.

[17] Princeton University. "WordNet." https://wordnet.princeton.edu (accessed: Nov. 11, 2020).

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in International conference on machine learning, 2014, pp. 1188-1196.

[19] M. Kadoguchi, S. Hayashi, M. Hashimoto, and A. Otsuka, "Exploring the Dark Web for Cyber Threat Intelligence using Machine Learning," in 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), 2019: IEEE, pp. 200-202.

[20] L. T. B. Ranera, G. A. Solano, and N. Oco, "Retrieval of Semantically Similar Philippine Supreme Court Case Decisions using Doc2Vec," in 2019 International Symposium on Multimedia and Communication Technology (ISMAC), 2019: IEEE, pp. 1-6.

[21] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, 2017.

[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," arXiv preprint arXiv:1607.01759, 2016.

[23] V. Zolotov and D. J. a. p. a. Kung, "Analysis and optimization of fasttext linear text classifier," 2017.
[24] I. Santos, N. Nedjah, and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network with fastText embeddings," in 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), 2017: IEEE, pp. 1-5.
[25] D. Gromann and T. Declerck, "Comparing pretrained multilingual word embeddings on an ontology alignment task," in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.
[26] S. Robertson and H. Zaragoza, The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc, 2009.
[27] kbandla. "Aptnotes." https://github.com/aptnotes/data (accessed: Nov. 11, 2020).

RISK ANALYSIS OF SETTING UP A RESTAURANT AT NYC

Santoshi Laxmi Reddy Ellanki[1] and John Jenq[2]

[1]University of Texas Medical Branch, Houston, Texas, USA
[2]Department of Computer Science, Montclair State University, NJ, USA

ABSTRACT

In this report, a system was developed that can predict the outcome of opening a restaurant in NYC based on various NYC open data sets, such as 311 calls, New York Police crime records and restaurant rating data. The data sets were preprocessed and cleaned before analysis to improve the quality of our results.

KEYWORDS

Big data, Risk analysis, PySpark, Decision tree.

1. INTRODUCTION

In today's highly competitive world, every business has a motive to be profitable. Among business sectors, restaurant's business is more connected to locality they are set up in. Apart from the quality of the food provided in the restaurant, there are other factors that need to be considered in order to make a restaurant business successful. People often choose restaurants that are in a safe and secure locations so they can relax and enjoy the meal. Some of the most important factors to consider regarding safety and accessibility are crime rate, entertainment, ease of commute and infrastructure. For example, an area with higher average income, and lower complaints of rodents, potholes, etc. can be considered as more safe and secure than other areas.

At an estimated population of 8.4 million, New York City is the most populated city in the United States. There is a constant fluctuation in the number of people moving to and from NYC every year, which has led to it having one of the most dynamic real estate markets worldwide. With such dynamic nature which will affect the locality rating, there is a need to constantly analyse the different localities of NYC. Someone moving into the city for setting up a business (e.g., restaurant) might be very interested in knowing which areas are more likely to bring in larger profit, which areas have good amenities, and less crime complaints. In this regard, metrics that show how safe a locality is, or which localities are prone to higher crime rate or higher service requests, are important factors to consider before setting up a restaurant. Financial institutes need to analyse risk when a business owner want to get finance from them. A tool to analyse risk would be beneficial.

Customer ratings for a restaurant are one of the important factors which contribute to the restaurant's success. If we take into consideration crime reports, 311 requests and the ratings of other restaurants in the same locality, the success of opening a new restaurant in that area can be predicted. These of course depend on the quality of the restaurant itself and how its facilities are. But we believe that the surroundings do affect the success of a restaurant. This is where the analytic comes in.

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 21-27, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110703

In this report, our goal is to come up with a dynamic model which can predict the outcome of opening a restaurant in NYC. The same can be applied to other cities in the US. To achieve this, we need to find relevant data and perform analysis on that data to define a dynamic and robust model. The advent of Big Data and advancements in computational techniques have enabled us to optimally find metrics that can help us to choose big data. We want the data set to be as substantial and complete as possible so that it covers a long history of transactions and information. The more comprehensive the data, the bigger the possibility of building an accurate model and producing meaningful results. The data sets we are using are from NYC open data website. This website will be used to import data sets for 311, NYPD and restaurant data. The data sets were comprehensive enough to suit our needs. [6, 7, 8].

Since there are only few technologies that can handle huge data, it is important that we choose a technology which is freeware or processed with a minimal cost. When processing large data, one machine with more memory and disk space is more expensive and less efficient than a group of machines with cheap hardware and configuration. As a result, we turned to big data technologies to make use of cluster machines and lightning-fast processing of data. After deciding to use big data technology to process the data, it is also important to choose the correct big data technology. We will choose a popular technology, Spark [5], which is built on Hadoop [4].

As Hadoop is the heart and soul of big data technologies to create and maintain a cluster, it is important to take advantage of Hadoop for cluster management and HDFS. We also need a technology to perform data operations, so we need to choose a platform that can handle huge data and perform operations. As Hive [11] is a popular data warehouse technology, we can use Hive for any data operations. These big data technologies are able to complete our data processing operations an efficient manner, so we decided to use them.

During the planning stages, in order for restaurant owners to accurately predict the success of opening a new restaurant, a few important factors need to be considered. These predictions would help the restaurant owner decide where to set up a restaurant based on location, crime and other factors to reduce the business risk. To make this prediction, we first need to identify patterns in the data so that the model can learn from these patterns and apply them to predict future behaviours and patterns. There are many algorithms that we can use to do this, and in this report, we used the simple decision tree algorithm [9, 12] because its similarity to human thinking and ease of use both result in good interpretations of the data. Some researchers even use this approach to identify risk factors for relapse to Smoking [2]. If we are clear on which tree nodes and attributes to choose, we can definitely produce a dynamic and robust model. Decision tree is a classification technique and the decision tree algorithm tries to solve the problem by using a tree representation. Each internal node of the tree corresponds to an attribute and each leaf node corresponds to a class label.

2. SYSTEM IMPLEMENTATION

Ubuntu [10] was chosen as the platform and Spark was installed on top of it. The data source includes 10 gigabytes of 311 service requests, where each record includes the type of call, latitude and longitude of the incident, zip code of the incident and the date of the complaint. The second data set is 1.4 gigabytes and includes New York City Police Department records reported crime and offense data based upon New York State Penal Law and other New York State Laws. Records specify type of crime, latitude and longitude of the incident, zip code of the crime scene and the date of the crime complaint. The third data set is 300 MB of restaurant rating records which includes restaurant zip code, building, street, and restaurant rating. Figure 1 shows the system components and their connections.

For these three major datasets, each contained millions of records. A substantial amount of time was spent on data processing and preparing the data sets for analysis. The data sets were cleaned using Hive SQL, where we handled missing values, duplicate values, records with invalid zip codes, records with invalid latitude and longitude information and determined the important fields to retain.

Some of the records were recorded at a latitude and longitude level instead of at a zip code level. Thus, we used the Geocode API and developed a python script that accepts latitude and longitude information as input parameters and outputs the corresponding zip code. Using Hive SQL, we joined the zip code of respective latitude and longitude location with the remaining attributes in the dataset. Finally, we used SQL and Hive to further process the fields which had a multitude of information contained in lists. These steps were taken in order to prepare individual datasets for the desired analytics. These refined individual datasets were grouped together using Hive SQL and SQL.

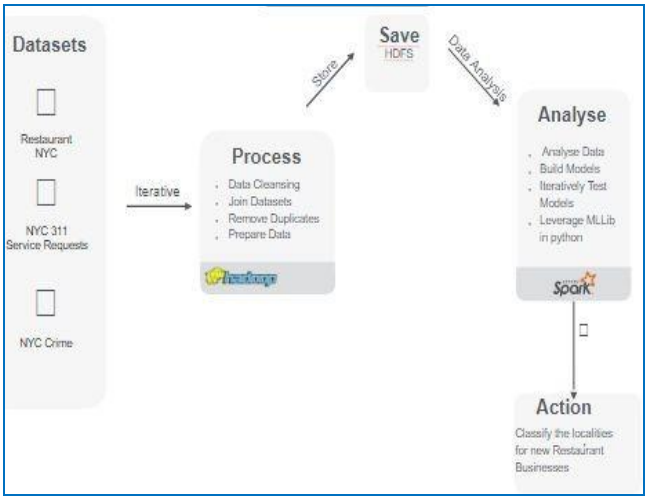


Figure 1. System components

2.1. Severity Classification Based on 311 Requests

We used 311 requests from past 5 years. Features that we considered after cleaning the data sets are 311_Incident Zip, which represent the 228 zip codes of NYC, and 311_Severity_1, 311_Severity_2 and 311_severity_3, which represent the count of 311 complaints based on severity. Complaints such as food poisoning and drug activity … etc., come under 311_Severity_1. Sidewalk condition, curb condition, mosquitoes …etc., come under 311_Severity_2 and complaints such as no permit parking, or lack of public bathroom come under 311_Severity_3. For each zip code, we calculated the count of all the severities and pivoted the severity counts as 311_Severity_1, 311_Severity_2, 311_Severity_3 for each zip code using SQL. See figure 2 for the classification.

311 Complaint severity	311 Complaint types
311_Severity_1	Food poisoning, Drug activity etc.
311_Severity_2	Sidewalk condition, curb condition, mosquitoes
311_Severity_3	No permit, public toilet

Figure 2. Classification of 311 Severity Based on 311 Complaint Types.

2.2. Severity Classification Based on Crime Complaints

We used crime complaints from past 5 years. Features that are considered after cleaning the data sets are Crime_Incident_Zip, which represent the 228 zip codes of NYC, and Crime_Severity_1, Crime_Severity_2 and Crime_severity_3, which represent the count of crime complaints based on severity. Crimes such as murder, felony … etc., come under Crime_severity_1. Forgery, robbery, assault … etc., come under Crime_severity_2 and the miscellaneous complaints come under Crime_severity_3. Similar to the 311 severity data sets, we calculated the count for each zip code with all the severities and pivoted the severity counts as Crime_Severity_1, Crime_Severity_2, and Crime_Severity_3 for each zip code using SQL. See Figure 3.

Crime Complaint severity	Crime Complaint types
Crime_Severity_1	Felony, murder, etc.
Crime_Severity_2	Assault, robbery, forgery
Crime_Severity_3	Miscellaneous

Figure 3. Classification of crime severity based on crime complaint types

2.3. Determine Restaurant Label Based on the Restaurant Rating

The features considered in the restaurant dataset are the Restaurant zip of the NYC locality, restaurant address, type of cuisine and the rating of the restaurant. The average of the ratings given to all the restaurants are considered. Restaurant rating which 2.5 or below is labelled as “low”, which is low recommended restaurant, Restaurant rating greater than 2.5 and less than 3.3 is labelled as “medium”, which is a medium recommended restaurant. Restaurant rating greater than 3.3 is labelled as “high”, which is highly recommended restaurant. See Figure 4 for restaurant labelling

Restaurant Label	Restaurant Rating
Low	2.5 or less
Medium	>2.5 and <3.3
High	3.3 and above

Figure 4. Restaurant Labelling Based on Rating

2.4. Determine the 311 Status, Crime Status Based on Crime Complaint Severity and 311 Complaint Severity

If the 311_Severity_1 is greater than 311_Severity_2 and 311_Severity_3, then 311_status is labelled as “low” and therefore is low recommended location. If the 311_Severity_2 is greater than 311_Severity_1 and 311_Severity_3 then the 311_status is labelled as “medium”, which is medium recommended location. Similarly, if the 311_Severity_3 is greater than 311_Severity_1 and 311_Severity_2 then the 311_status is labelled as “high” and this means the area is highly recommended. See Figure 5 for 311_status classification.

We classified crime severity in a similar way. If the Crime_Severity_1 is greater than both Crime_Severiy_2 and Crime_Severity_3, then Crime status is marked as “low” which means it is not a recommended location. If the Crime_Severity_2 is greater than both Crime_Severity_1 and Crime_Severity_3, then Crime status is labelled as “medium” and it is medium recommended location. If 311_Severity_3 is greater than both 311_Severity_2 and 311_Severity_1, then the

Crime status is labelled as “high” and it is a highly recommended location. Figure 6 shows the classification of crime status based on the crime complaint severity.

311 Complaint severity	311 status
311_Severity_1>311_Severity_2 & 311_Severity_1>311_Severity_3	Low
311_Severity_2>311_Severity_1 & 311_Severity_2>311_Severity_3	Medium
311_Severity_3 > 311_Severity_1 & 311_Severity_3 > 311_Severity_2	High

Figure 5. The 311 Status classification

Crime Complaint severity	Crime status
Crime_Severity_1 > Crime_Severity_2 & Crime_Severity_1 > Crime_Severity_3	Low
Crime_Severity_2 > Crime_Severity_1 & Crime_Severity_2 > Crime_Severity_3	Medium
Crime_Severity_3 > Crime_Severity_1 & Crime_Severity_3 > Crime_Severity_2	High

Figure 6. Crime Status classification

2.5. Determine the Location Label Based on Crime Status, 311 Status and Restaurant rating

Looking at all the possible combinations of crime status, 311 status and restaurant rating, we labelled each area as best, moderate or worst. For instance, if the 311 status is low, crime status has a low classification and restaurant rating is less than 2.5, then we label this location as worst, which means the location is not recommended for setting up a restaurant. See Figure 7 for classifications.

311 status	Crime status	Restaurant Rating	Location label
Low	Low	<2.5	Worst
Low	Medium	<2.5	Worst
Medium	Low	<2.5	Worst
Medium	Medium	<2.5	Worst
High	Low	<2.5	Worst
High	Medium	<2.5	Worst
High	Low	2.5-3.3	Worst
Medium	Low	2.5-3.3	Worst
Low	Low	2.5-3.3	Worst
Medium	Medium	2.5-3.3	Moderate
Low	Low	>=3.3	Moderate
Medium	Low	>=3.3	Moderate
Low	Medium	>=3.3	Moderate
High	Medium	2.5-3.3	Moderate
High	Medium	>=3.3	Moderate
High	High	>=3.3	Best
High	Low	>=3.3	Best
High	Medium	>=3.3	Best
High	Medium	>=3.3	Best

Figure 7. Location Classification

2.6. Prediction Model Implementation

The Apache Spark MlLib was used to develop the prediction models. The feature set was as described above. We experimented with two different Machine Learning algorithms: Logistic Regression and Decision Trees. Logistic Regression is best suited for models with output ranging from 0 to 1. In this model, output was categorized as low, medium, or high ranging from 0 to 2. The dataset obtained after cleaning is split randomly by using the random function into both training and the testing datasets, known as K-fold cross validation approach. The model after training the dataset is applied to predict the results for the testing and the results are evaluated. We found that compared to decision trees, Logistic regression was a less accurate algorithm.

3. CONCLUSIONS AND REMARKS

The total count for prediction with low recommended is 4149036. For medium recommended, total count is 3984406 and for high recommended, 3273156. See Figure 8 for a pie chart illustrate the percentage of the recommendation. Due to the communication delays among cluster PCs the speed up is not significant.

Decision Tree Algorithm is used as a classification technique which is used on the existing dataset and the resulting decision tree is used to create a strategy for finding the status of the zip code based on the security and severity. The decision tree considers the factors that are considered relevant for the decision.

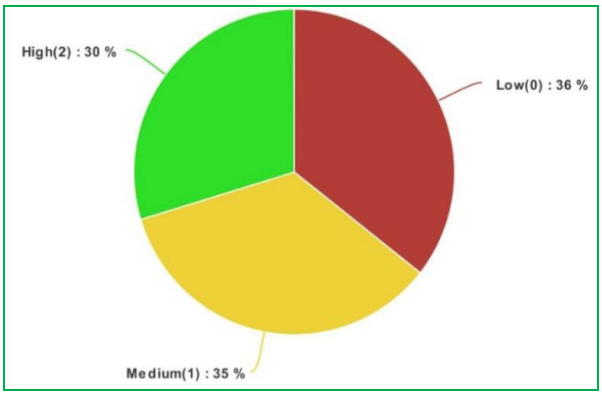


Figure 8. The Percentage of Recommendations for Low, Medium and High Recommendation ZIP Codes for Setting up a Restaurant.

The decision tree technique can be used to identify the impact of changes on results when one of the underlying attributes is changed. As the security and severity for a location changes constantly, this technique allows businesses to identify the factors that are more sensitive and less sensitive contributing to the security of the location. This kind of sensitivity is difficult to detect in another model.

4. FUTURE WORKS

Because there are many different algorithms and topics to explore in model creation and feature selection, it would be interesting to analyse the effects of these algorithms and verify if any of them could perform better on the data sets that we have.

Additionally, we can perform more robust analysis and develop a model with more accuracy if we have the ability to input additional information like housing rates, population demographics, accessibility to public transit, green space, school rating and population diversity. Due to time constraints, as well as the limit on resources and availability of public data, our analysis was restricted.

As the work is done on a local pseudo-distributed mode cluster and there is no significant speed up, Using a bigger cluster with more data on cloud computing service like Amazon Web Service maybe a good option, which will give an opportunity to use more big data technologies and use of lightning-fast speed of the cluster.

REFERENCES

[1] Nitish Gupta, Sameer Singh, Collectively Embedding Multi-Relational Data for Predicting User Preferences https://arxiv.org/pdf/1504.06165.pdf
[2] Using decision tree analysis to identify risk factors for relapse to Smoking, by Megan E. Piper, Wei-Yin Loh, Stevens S. Smith, Sandra J. Japuntich, Timothy B. Baker, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2908723/pdf/nihms176002.pdf
[3] Scott L. Minkoff, NYC 311: A Tract-Level Analysis of Citizen-Government Contacting in New York City, https://www.researchgate.net/publication/274708724_NYC_311_A_Tract-Level_Analysis_of_Citizen-Government_Contacting_in_New_York_City
[4] Apache Hadoop http://hadoop.apache.org/
[5] Apache Spark http://spark.apache.org/
[6] NYC 311 Calls Data https://data.cityofnewyork.us/dataset/311-Service-Requests-From-2011/fpz8-jqf4
[7] NYC Crime Reports Data https://data.cityofnewyork.us/Public-Safety/Historical-New-York-City-Crime-Data/hqhv-9zeg
[8] NYC Restaurant Data https://www.yelp.com/dataset
[9] Introduction to Data Mining, Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. Pearson Publishing
[10] Ubuntu Download resource https://www.ubuntu.com/download/desktop
[11] Hive https://hive.apache.org/
[12] Decision tree analysis for the risk averse organization. Hulett, D. T. Paper presented at PMI® Global Congress 2006—EMEA, Madrid, Spain. Newtown Square, PA: Project Management Institute. Also https://www.pmi.org/learning/library/decision-tree-analysis-expected-utility-8214

INFORMATION TECHNOLOGY GOVERNANCE OF JAPANESE COMPANIES; AN EMPIRICAL STUDY

Michiko Miyamoto

Department of Management Science and Engineering,
Akita Prefectural University, Yurihonjo City Akita, Japan

ABSTRACT

IT has become an essential part of the organization. IT governance specifies the decision rights and accountability framework to encourage desirable behaviour in using IT. Concepts of IT governance has expanded to improve IT-business alignment under today's business environment and prospects. This paper contributes to empirically knowledge of IT governance practices in Japanese organizations based on survey data gathered from 101 corporations, including large, medium, and small companies. The findings of the ordinal regression analyses in this study indicate that IT governance is associated with Strategic Alignment, Performance Measurement and Value Delivery, while Risk Management and Resource Management have positive but no significance association with IT governance.

KEYWORDS

IT Governance, IT-Business Alignment, Strategic Alignment Maturity, Regression Analysis.

1. INTRODUCTION

According to the "Corporate IT Trend Survey 2020" (FY2019 survey) by Japan Users Association of Information Systems [1] , which examines trends in corporate IT investment and IT strategy, 40.7% of the total answered their IT budgets would "increase", 46.0% "unchanged", and 13.2% "decrease." Digital transformation in business continues to be an important management issue. The role of IT continues to grow, and IT budgets are expected to continue their upward trend despite growing economic uncertainty.

Today, IT has become an essential part of the organization, and IT function of the organization has evolved from a technology provider into a strategic partner [2]. IT governance addresses the authority and control for key IT activities in organizations, such as IT infrastructure and IT use. Effective IT governance guarantees that IT helps business goals, maximizes investment in IT, and directs IT-related risks and opportunities [3]. A lack of IT governance in one company can threat an entire society, as shown by two well-publicized IT failures in Japan: Japan Airlines' halting automated check-in procedures by the system failure, and Mizuho Financial Group's banking system failure. Those system failures caused chaos, from which the recovery took for a while until the operations went completely back to normal.

Weill and Ross [4] define IT governance as specifying the decision rights and accountability framework to encourage desirable behaviour in using IT. In corporate management, IT governance, which is an organizational mechanism for continuously optimizing IT investment,

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 29-40, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110704

effects, and risks, can be the most important issue for Japanese companies. IT governance influences corporate management strategy. Concepts of IT governance has expanded [3][4] to improve IT-business alignment under today's business environment and prospects.

This paper contributes to the empirical knowledge of IT governance practices in Japanese organizations by using a questionnaire survey data. Factor analyses and multiple regression have been used to interpret the relationships among IT governance and IT governance related factors.

The remainder of this paper is organized as follows. Section 2 presents a literature review of the concept of IT governance, IT governance framework, strategic alignment maturity, and IT governance in Japan. Section 3 describes research objectives and hypotheses. Section 4 presents the methodology used for questionnaire survey and multivariate statistical approach. Section 5 presents our survey results and implications. The conclusion is presented in Section 6.

2. LITERATURE REVIEW

IT governance takes place through the specification of decision rights and accountabilities framework designed to motivate advantageous IT-related behaviour within an organization [4]. IT management must ensure the governance mechanisms are in place implemented to fulfil the strategies [5]. Many literatures have conducted a systematic literature review on IT governance and listed various definitions (e.g. [6] [7] [8] [9] [10]).

Carroll, et al [6] focused on the published literature on the control objective for information and related technology (COBIT). COBIT and "value from IT investments" (Val IT) are frameworks developed by Information Systems Audit and Control Association (ISACA) for information technology (IT) management and IT governance to help business executives, IT personnel and management. They found approximately 7 % of the publications had academic background, while 93% were practitioners oriented. Even the later study by De Maere and De Haes [11] suggested the number of publications in top journals is low at 1 % within the context of IT governance.

Based on their literature review, Vejseli and Rossmann [8] identified five relevant perspectives for further research including strategic alignment perspective, IT leadership perspective, IT capability and process performance perspective, resource relatedness perspective and culture perspective. Fink and Ploder [12] introduced a decision support framework to address the issue of implementing IT governance into the organizational context and imply corporate culture can influence the success of IT governance implementation. Their model is developed based on the IT governance model suggested by Weil and Ross [4] with its five decision fields: IT principles, IT architecture, IT infrastructure, business applications, IT investment and prioritization. Webb et al. [7] proposed definitive definition of IT governance, focusing on two important areas of influence on the emergence of IT governance: 1) corporate governance within organization and 2) strategic information system. They found five elements (strategic alignment, delivery of business value through IT, performance management, risk management and control and accountability) of the framework have been validated.

Aasi et al. [10] found that IT governance is strongly linked with the corporate governance, which is influenced by culture; however, there are few research studies in this topic. After performing an extensive literature review, Levstek et al. [9] decided to use the following definition of De Haes and Van Grembergen [13] as the most comprehensive definition.

"IT governance is an integral part of corporate governance, exercised by the Board, overseeing the definition and implementation of processes, structures and relational mechanism in the

organization that enable both business and IT people to execute their responsibilities in support of business/IT alignment and the creation of business value from IT enabled business investment" (see Figure 1).

Symons et al. [14] suggested that implementing a good IT governance requires a framework based on structure, process, and communication. Symons et al. [14] also suggests that there are four objectives that drive IT governance: IT value and alignment, accountability, performance measurement, and risk management. Each of these objectives must be addressed as part of the IT governance process (see Figure 2).

There are many theoretical literatures, discussing over different definitions of IT governance and frameworks, while fewer empirical studies are found.

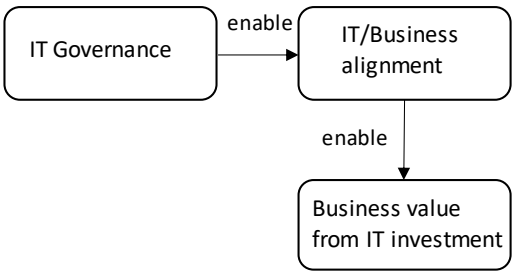


Figure 1. The definition of IT Governance
(Source: De Haes & Van Gembergen [11])

Lunardi et al. [15] empirically tested the determinants for the effectiveness of IT governance, based on survey data of 87 CIOs of large Brazilian companies. They found that IT strategic alignment, IT value delivery, IT risk management and IT performance management are positive and significant associated to IT governance effectiveness, indicating that the higher the performance of these domains, the higher the IT governance effectiveness. IT strategic alignment, in turn, appears as the main predictor. Miyamoto and Kudo [16] conducted an empirical research on IT governance using research data of 345 SMEs residing in Akita prefecture, located in the northern part of Japan. They found that Strategic Alignment, Performance Measurement, Resource Management, Risk Management, Value Delivery are positive and significant associated to IT governance. They also found that most of these SMEs understand the importance of IT governance, but few have a person in charge of IT specialization.

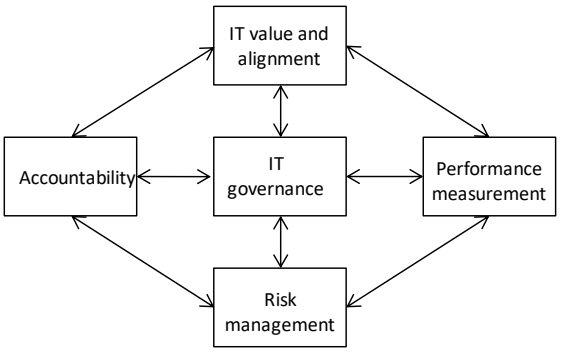


Figure 2. Four objectives that drive IT governance [14]

3. RESEARCH MODEL AND HYPOTHESES

The proper alignment between use of IT and the business goal of an organization is basic principal to efficient and effective IT governance [6]. Based on literature review, the author assesses Japanese corporations' IT governance by strategic alignment, risk management, value delivery, resource management, and performance management, and created a research model (see Figure 3).

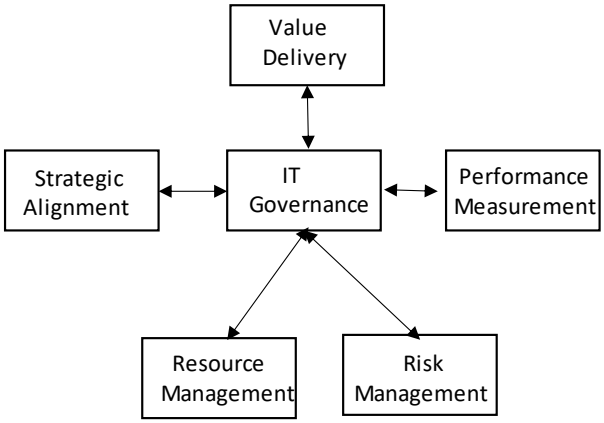


Figure 3. Research model

The following hypotheses are proposed and are examined.

H1: There is a significant, positive relationship between IT governance and Strategic Alignment.
H2: There is a significant, positive relationship between IT governance and Performance Measurement.
H3: There is a significant, positive relationship between IT governance and Resource Management.
H4: There is a significant, positive relationship between IT governance and Risk Management.
H5: There is a significant, positive relationship between IT governance and Value Delivery.

4. SURVEYS

4.1. Data

The data were collected by means of a questionnaire. The survey was conducted throughout Japan from late August in 2018 to mid-November in 2018. Respondents were randomly selected from several databases of local businesses of each prefecture and from members of Japan Users Association of Information Systems. The survey was conducted online and amassed 101 valid responses. The questionnaire was sent by email to the information system division, as well as the corporate planning division of the firms. Most of the questionnaires are asked by 5 -point scale.

Table 1. Size of the companies studied

Total number of employees		Annual Revenue (in billion yen)	
More than 1,000	27	More than 50	25
300-1,000	24	30 - 50	10
100-300	14	10 - 30	14
50-100	10	5 - 10	4
20-50	13	0.5 - 5	27
Less than 20	12	Less than 0.5	18
Missing	1	Missing	3
Total	101	Total	101

Table 1 shows number of employees and annual revenue of the sample. The data in this study contains both large corporations as well as SMEs. In Japan, SMEs is defined under Article 2, Paragraph 1 of the Small and Medium-sized Enterprise Basic Act, and the term “small enterprises” refers to “small enterprises” as defined under Article 2, Paragraph 5 of said act. The category of “more than 1,000 employees” and “300 to 1,000 employees” can be considered as large corporations: those of up to 300 employees as small and medium enterprises, and those less than 20 employees as small enterprises.

The respondents in this study represented a variety of industry. The list of industries for those participating this research is shown in Table 2 and the list of variables is shown in Table 3.

Table 2. The Participating Organizations: Industries

Industries	Frequencies	%
Rubber / ceramic industry	2	2.0
Other service industry	8	7.9
Other manufacturing industry	10	9.9
Chemical · Petroleum	1	1.0
Machinery and electrical equipment	14	13.9
Eucation	2	2.0
Finance / insurance industry	1	1.0
Construction	8	7.9
Trading company · other wholesale business	8	7.9
Retail · Food industry	5	5.0
Telecommunications	24	23.8
Food	3	3.0
Fisheries · Agriculture · Forestry · Mining	1	1.0
Precision equipment	2	2.0
Fiber	2	2.0
Warehouse · Transportation	1	1.0
Steel · Nonferrous metal · Metal	2	2.0
Electricity, Gas and Public Service	2	2.0
Real estate business	2	2.0
Transportation equipment	3	3.0
Total	101	100.0

4.2. Empirical Analyses

The ordinal regression model [17] is used to model between the ordinal outcome and independent variables. Both dependent variables (IT governance (ITG) and independent variables (Strategic Alignment (SG), Risk Management (RM), Value Delivery (VD), Resource Management (RM), Performance Management (PM) are obtained by confirmatory factor analysis.

4.2.1. Confirmatory Factor Analysis

In the social sciences, factor analytical methods are commonly used in the scale measurement in examining the structure of scales. There are two common standard statistical tools, exploratory and confirmatory factor analyses for developing measurement scales.

Confirmatory factor analysis (CFA) is a multivariate statistical procedure that is used to test how well the measured variables represent the number of constructs. In CFA, the number of factors requires in the data can be specified, and measured variable is related to which latent variable.

The basic factor analysis equation can be represented in matrix form as:

$$Z = \lambda F + \varepsilon$$

Where Z is a px1 vector of variables, λ is a pxm matrix of factor loadings, F is an mx1 vector of factors and ε is a px1 vector of error or residual (unique or specific) factors [18]. Because of differences in the units of variables used in factor analysis, the variables were standardized, and a correlation matrix of variables was used to obtain eigenvalues. Varimax rotation was used to facilitate interpretation of factor loadings (L_{ik}). Factor coefficients (C_{ik}) were used to obtain factor scores for selected factors. Factors with eigenvalues greater than 1 were employed in multiple regression analysis [18] [19] [20]. Score values of selected factors were considered as independent variables for predicting IT governance.

Table 3. A List of Variables

	Variables	
Strategic Alignment	importance	a) Importance of leveraging IT for enhancing competitiveness
	top1	c) Involvement of IT personnel to process business strategy.
	top2	d) Involvement of senior management to IT strategy formulation process
	top3	e) Involvement of senior management for business transformation projects involving IT
	top4	f) Aggressiveness of management for communication with IT personnel
	top5	g) Senior management's IT strategy is known to every employee.
	top6	h) Management suggests and supports utilization of IT in business
Risk Management	risk1	a) Managers communicate with IT personals to understand the circumstances of their use of IT
	risk2	b) IT personnel substantively understand the management strategy
	risk3	c) The business unit personnel understand IT environment and the company's IT strategy
	risk4	d) Exchanging ideas between departments by leveraging information sharing and corporate intranet groupware
	risk5	e) Hold regular meetings on IT projects
IT governance	gove1	a) Supervision and management of IT budget
	gove2	b) Supervision and management of IT investment evaluation
	gove3	c) Thorough sharing duties and authority on IT
	gove4	d) Establishment of the IT-related committee
	gove5	e) Clarification of the criteria in an allocation and prioritization of IT utilized resources
	gove6	f) Standardization of IT adoption process
	gove7	g) IT security risk management and supervision
Value Delivery	value1	a) Understanding the business value expected in the use of IT between business sectors
	value2	b) Participation in the implementation process of IT employees
	value3	c) Participation in the management planning process of IT personnel
	value4	d) Each department and senior management to share each other's goals and risk
	value5	e) Business divisions and IT personnel trust each other
	value6	f) Regarding IT projects, consult professionals (such as the IT coordinator) or external organizations, such as public institutions and private companies.
Performance Management	skill1	a) Encourage and offer a chance to take advantage of in-house IT employees to create new ways to use IT
	skill2	b) Educate and train to increase the capacity utilization of IT
	skill3	c) Set a goal of IT skills of employees and recommend employees to take the IT related exam
	skill4	d) Hiring personnel with the knowledge and skills required for the IT management and operation
	skill5	e) Creating managerial posts for IT professionals
	skill6	f) Expanding the career paths for IT professionals
Resource Management	resource1	a) Computerize general administrative duties (e.g., planning, finance, accounting, regulatory measures, and quality control)
	resource2	b) Computerize personnel and labor management (e.g., human resources management and benefits, recruitment and training of personnel, salaries payments, etc.)
	resource3	c) Computerize technological development (e.g., R & D, product design, knowledge management, and production equipment design)
	resource4	d) Computerize procurement (e.g., demand planning, payment and billing, procurement, etc.)
	resource5	e) Computerize purchasing and logistics (e.g., scheduling, shipment and delivery planning, warehouse management, inventory management, etc.)
	resource6	f) Computerize manufacturing operations (e.g., assembly, maintenance, equipment, equipment maintenance, inspection, printing, etc.)
	resource7	g) Computerize logistics shipping (e.g., order processing, shipping and transportation planning of the final product, and storage of the final product)
	resource8	h) Computerize marketing and sales (e.g., advertising, sales, promotion, etc.)
	resource9	i) Computerize servicing (e.g., maintenance and repair of the final product, management and customer support)

Table 4. A Correlations Matrix

	Strategic Alignment	Risk Management	IT Governance	Value Delivery	Resource Management	Performance Management
Strategic Alignment	1	.818**	.775**	.743**	.334**	.666**
Risk Management	.818**	1	.786**	.759**	.417**	.698**
IT Governance	.775**	.786**	1	.830**	.419**	.729**
Value Delivery	.743**	.759**	.830**	1	.338**	.613**
Resource Management	.334**	.417**	.419**	.338**	1	.475**
Performance Management	.666**	.698**	.729**	.613**	.475**	1

**. Correlation is significant at the 0.01 level (2-tailed).

Table 4 contains the Pearson correlation coefficient between all pairs of six latent variables obtained by factor analysis. Correlation is significant at the 0.01 level (two-tailed) and the correlations between indicators range between 0.33 and 0.83. All variables are correlated with other variables well, but none of the correlation coefficients is particularly larger than 0.90; therefore, multicollinearity is not a problem for these data.

4.2.2. Regression Analysis

The regression equation fitted was:

$$ITG_{ij} = a + b_1SA + b_2RM + b_3VD + b_4RM + b_5PM + e$$

Where *a*, is regression constant (it is the value of intercept and its value is zero); b_1,…b_5, are regression coefficients of Factor Scores (FS), and *e* is the error term. Regression coefficients were tested with a t-statistic. The coefficient of determination (R2) was used as an indicator of the quality of the regression [21].

5. EMPIRICAL RESULTS

The results of analyses are shown as follows.

Table 5. Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.880[a]	.775	.762	.44616271

The table 5 is the model summary. R can be considered as one measure of the quality of the prediction of the dependent variable, IT governance. A value of 0.880 indicates a good level of prediction. The R square (R^2) value, the coefficient of determination, shows the value of 0.775, which indicates the independent variables explain 77.5% of the variability of the dependent variable.

Table 6. ANOVA

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	62.932	5	12.586	63.229	.000[b]
	Residual	18.314	92	.199		
	Total	81.245	97			

The F-ratio in the ANOVA (Table 6) tests whether the overall regression model is a good fit for the data. The table shows that the independent variables statistically significantly predict the dependent variable, F (5, 92) =63.229, p (.001) < 0.05 (*i.e.*, the regression model is a good fit of the data).

Table 7 shows the results of regression analyses.

Table 7. Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.001	.045		-.016	.987
	Strategic_Alignment	.149	.087	.153	1.719	.089
	Risk_Management	.116	.105	.106	1.103	.273
	Value_Delivery	.438	.077	.442	5.716	.000
	Resource_Management	.068	.056	.071	1.203	.232
	Performance_Management	.254	.074	.254	3.433	.001
a. Dependent Variable: IT_Governance						

Value Delivery *p* (0.000) <0.05, Performance Management *p* (0.001) < 0.05, and Strategic Alignment *p* (0.089) < 0.10 are statistically significant, but Risk Management *p* (0.273) > 0.05 and Resource Management *p* (0.232) > 0.05 are not significant.

A result of IT governance model for Japanese SMEs shows the following findings.

H1: There is a significant, positive relationship between IT governance and Strategic Alignment.
H2: There is a significant, positive relationship between IT governance and Performance Measurement.
H3: There is positive, but no significant relationships are found between IT governance and Resource Management.
H4: There is positive, but no significant relationships are found between IT governance and Risk Management.
H5: There is a significant, positive relationship between IT governance and Value Delivery.

The results suggest that all hypotheses show positive relationships with IT governance; particularly those with Strategic Alignment, Performance Measurement and Value Delivery are positive and statistically significant.

6. CONCLUSIONS

Based on literature review on IT governance, the author has tested factors which are related to IT governance, using the survey data from 101 Japanese corporations including large ones and SMEs.

The findings of the ordinal regression analyses in this study indicate that IT governance is associated with Strategic Alignment, Performance Measurement and Value Delivery, while Risk

Management and Resource Management have positive but no significance association with IT governance.

The results suggest that the corporations recognize that IT strategic alignment as an important component to attain higher levels of IT effectiveness, which in turn could help organizations to obtain better business performance (Luftman at al. 2010). The respondents also seem to understand that measuring IT performance and value delivery are essential parts of any IT governance program.

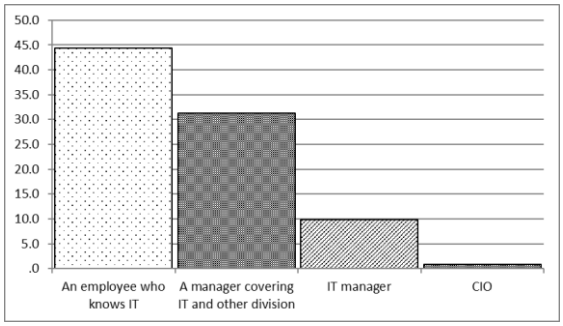


Figure 3. Who is responsible for IT in 2013 [16]

Figure 4. Who is responsible for IT in 2018

However, the author could not find significant associations between IT governance and IT risk and resource management. Although many organizations increasingly recognize the importance of IT governance, it is also known that many of them struggle with implementing and embedding these governance practices into their organization [22].

Results for a question asked, “Who is responsible for IT in your organization?” are shown in figure 3 and figure 4, respectively. Figure 3 reflects the results of questionnaire targeting SMEs in Akita prefecture, the northern part of Japan, conducted in 2013, and figure 4 reflects those of entire Japan, including large companies and SMEs, in 2018. Figure 3 suggests that “an employee who knows IT is responsible for IT” is more than 50 percent, and there are a few CIOs. The result for SMEs in 2018 seems to have the same trend; however, larger corporations have more dedicated IT managers, including CIOs.

The limitation of this study is “sample bias.” About one fourth (24%) of respondents are representing the telecommunications industry, and 14% are those of the machinery and electrical equipment, 10% are representing other manufacturing industry (see Table 2). This paper might have had limited ability to gain access to the appropriate scope of participants. The author is planning to conduct a larger survey research to solve such bias.

ACKNOWLEDGEMENTS

This work was supported in part by JSPS Grants-in-Aid for Scientific research (C) Number JP20K01853.

REFERENCES

[1] Japan Users Association of Information Systems, (2020) “Corporate IT Trend Survey 2020 (FY2019 survey)”, Retrieved June 11 2020 from https://juas.or.jp/library /research_rpt/it_trend/.
[2] HP, (2003) “HP IT Service Management (ITSM), Transforming IT organizations into service providers”. Retrieved June 10, 2020 from https://docplayer.net/19185869-Hp-it-service-management-itsm.html.
[3] Schwalbe, Kathy, (2010) *Information Technology Project Management*, Revised, 6th Edition, Cuorse Technology.
[4] Weill, Peter David & Ross, Jeanne W., (2004) “IT governance: How Top Performers Manage IT Decision Rights for Superior Results”, *Harvard Business School*, Boston.
[5] Wilbanks, Linda (2008) “IT Management and Governance in Equal Parts”, *IT Professional*, vol. 10, no. 1, Jan./Feb., pp.60-61.
[6] Carroll, Peter, Ridley, Gail & Young, Judy (2004) “COBIT and its utilization: a framework from the literature”, *System Sciences*, January, pp.233-240.
[7] Webb, Phyl, Pollard, Carol & Ridley, Gail (2006) “Attempting to Define IT Governance: Wisdom or Folly?” *Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS) Vol. 8*, ISBN: 0-7695-2507-5, 4-7, January 2006, Hawaii, USA, 194a.-194.1.
[8] Vejseli, Sulejman & Rossmann, Alexander (2017) “The Impact of IT Governance on Firm Performance A Literature Review”, *PACIS 2017 Proceedings*. 41.
[9] Levstek, Aleš, Hovelja, Tomaž & Pucihar, Andreja (2018) "IT Governance Mechanisms and Contingency Factors: Towards an Adaptive IT Governance Model", *Organizacija* 51(4), pp.286-310.
[10] Aasi, Parisa, Rusu, Lazar & Vieru, Dragos (2017) "The Role of Culture in IT Governance Five Focus Areas: A Literature Review", *International Journal of IT/Business Alignment and Governance (IJITBAG)*, 8(2), pp.42-61.
[11] De Maere, Koen & De Haes, Steven (2017) “Is the Design Science Approach fit for IT Governance Research?”, *ECRM 2017 16th European Conference on Research Methods in Business and Management Studies*, pp.399-407.
[12] Fink, Kerstin & Ploder, Christian (2008) “Decision Support Framework for the Implementation of IT-Governance”, *in Proceedings of the 41st Hawaii International Conference on System Sciences*, Los Alamitos, CA: IEEE, Paper 432. 2008.
[13] De Haes, Steven & Van Grembergen, Wim (2009) “An Exploratory Study into IT Governance Implementations and its Impact on Business/IT Alignment”, Information Systems Management, 26, (2), pp.123-137.
[14] Symons, Craig (2005) *IT Governance Framework*, Forrester Research, Inc.
[15] Lunardi, Guilherme Lerch, Maçada, Antonio Carlos, & Becker, João Luiz (2017) “Antecedents of IT Governance Effectiveness: An Empirical Examination in Brazilian Firms”, *Journal of Information Systems* 31 (1), pp.41–57.
[16] Miyamoto, Michiko & Kudo, Shuhei (2013) “Five Domains of Information Technology Governance in Japanese SMEs; An Empirical Study”, *International Conference on ICT Convergence 2013* (ICTC2013), Proceedings, pp.964-969.
[17] McCullagh, Peter & Nelder, John A. (1983) Generalized linear models. Second ed. London: Chapman and Hall.
[18] Sharma, Subhash & Mukherjee, Soumen (1996) *Applied Multivariate Techniques*, John Wiley and Sons, Inc., New York.
[19] Tabachnick, Barbara G. & Fidell, Linda S. (2001) *Using Multivariate Statistics*. Allyn and Bacon A Pearson Education Company Boston.
[20] Johnson, Richard A. & Wichern, Dean W. (2002) *Applied Multivariate Statistical Analysis*. Prentice Hall, upper Saddle River, New Jersey.
[21] Draper, Norman Richard & Smith, Harry (1998) *Applied Regression Analysis* (Wiley Series in Probability and Statistics), Third Edition, Wiley.

[22] De Haes, Steven, Van Grembergen, Wim & Debreceny, Roger S. (2013) "COBIT5 and enterprise governance of information technology: Building blocks and research opportunities." Journal of Information Systems; 27(1): pp. 307-324.

AUTHOR

Michiko Miyamoto studied at the State University of New York College at Buffalo, where she received her Bachelor of Science degree (magna cum laude). She received her MBA from the University of California at Los Angeles. After a 7-year career with Goldman Sachs and Company, obtained her PhD further to a thesis about Econometrical Approaches to Economic and Strategic Management Studies at the University of Tsukuba, Graduate School of Systems Management. In 2008, she joined the Department of Management Science and Engineering at the Akita Prefectural University.

AUTOMATIC DETECTION AND EXTRACTION OF LUNGS CANCER NODULES USING CONNECTED COMPONENTS LABELING AND DISTANCE MEASURE BASED CLASSIFICATION

Mamdouh Monif[1], Kinan Mansour[2], Waad Ammar[2] and Maan Ammar[1]

[1]AL Andalus University for Medical Sciences,
Faculty of Biomed. Eng.,Al Qudmos, Syria
[2]Al Andalus University Hospital, Al Qudmos, Syria

ABSTRACT

We introduce in this paper a method for reliable automatic extraction of lung area from CT chest images with a wide variety of lungs image shapes by using Connected Components Labeling (CCL) technique with some morphological operations. The paper introduces also a method using the CCL technique with distance measure based classification for the efficient detection of lungs nodules from extracted lung area. We further tested our complete detection and extraction approach using a performance consistency check by applying it to lungs CT images of healthy persons (contain no nodules). The experimental results have shown that the performance of the method in all stages is high.

KEYWORDS

lungs cancer, lungs area extraction, nodules detection, distance measure, performance consistency check.

1. INTRODUCTION

According to World Health Organization (WHO) statistics, lung cancer cases exceeded 13% of all cancer cases appeared in the world, surpassing breast cancer which came second with 11.9%[1].These facts made lung cancer a major concern for both related specialists and scientists seeking efficient computer aided diagnosis.

Early diagnosis can improve the effectiveness of treatment and increase the patient's chance of survival [2]. The previous facts motivated researchers to pay a great attention to researches that work on automated diagnosis of lung cancer in a wide field known as Computer Aided Diagnosis Systems for Lung Cancer. A reliable computer diagnosis of the disease will help screening a large number of images created every day enabling specialized doctors to work with only little amount of candidate images and raising their efficiency [3]. Hundreds of published scientific papers appeared during the past 3 decades [2].These published research works were reviewed in several review papers [2,3,4,5,6] to evaluate the overall situation of research on this subject, identify the challenges, and propose specified points to improve the performance of the CAD approaches for lung cancer detection and diagnosis. The review papers discussed in general the four main steps of processing of lungs images: segmentation of the lung fields (regions),

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 41-53, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110705

detection of nodules inside the lung fields, segmentation of the detected nodules, and diagnosis of the nodules as benign or malignant.

In general, all review papers called for further improvements in the performance of the available systems, especially, in segmentation of lunges (lungs from chest image) and detection of lung nodules (segmenting and detecting nodules from lung images) and considered this improvement as challenges for further investigation in this field.

In this paper, we introduce an automatic method for detection and extraction of Lungs Cancer Nodules from chest CT images. The method provides good improvements in segmenting lungs accurately from chest CT image with a considerable variety of shapes so that it can be considered as flexible and effective. The method uses 2D Connected Components Labeling (2D-CCL) technique to extract the lungs area from chest image, and after extraction of some suitable features, the method makes use of a weighted distance measure based classification technique to detect nodules with high accuracy.

2. RELATED WORKS

Different techniques were used by researchers to extract nodules from 2D and 3D, CT images. Using two dimensional CT images, Kaur R., et al. [7] used PCA (Principal Component analysis) to extract nodules from lung cancer CT images, and Miwa T., et al.[8] used morphological N-Quoit Filter to automatically extract nodules based on shape and gray level information. Homma N., et al. [9] used Gabor filter and the difference of pixel values along the object axis to detect nodules, and Gomathi M., et al. [10] used FPCM and extreme learning machine for the same purpose. Those were some sample references from the period (2002-2013). Recently, S. Makaju et al. [11], used in 2018 watershed technique for segmentation and some shape and density features to detect nodules., S. Wang et. al. [12]used in (2020) Residual Neural Networks and N. Khehrah et. al [13] used in (2020) the histogram and some morphological operators to extract the lung, and a threshold based technique to select candidate nodules. The works mentioned above on detecting lung nodules from 2D chest CT images for Computer Aided diagnosis are naturally not exhaustive but give a good idea about the diversity of techniques and methods used.

Ammar M. et al. [14], used the CCL technique to extract liver area from the complicated 2D abdominal CT image to be used for diagnosis of liver cancer. Based on this experience, the authors explored the possibility of using the same technique for the detection and extraction of lung cancer nodules from 2D chest CT images, and presented the encouraging results they obtained in this paper.

3. USED DATA

Images from CT scans of lungs of 11 persons were provided by Alsham Imaging Center and 102 others by Tishreen Hospital. The images in each scan are about 80, and the thickness of each slice is 2 mm. The specialist selected one image from each scan to be used in this study. We divided the 113 images into 2 groups: (1) lungs of 98 cancer patients containing nodules, and (2) lungs of the remaining 15 persons with no nodules (from healthy persons). Both groups were used for lung area extraction from the CT image. For nodule detection, we used the first group for developing and testing the algorithm, and used the second group to check the consistency of the algorithm performance, since the algorithm that detects nodules in the lungs of cancer patients must detect "no nodules" in the images of the lungs of healthy persons. We transformed all the images from DICOM format to JPG format for processing in MATLAB environment. Fig. 1

shows an example from the CT slices of healthy persons, and another one from CT slices of a cancer patient lungs.

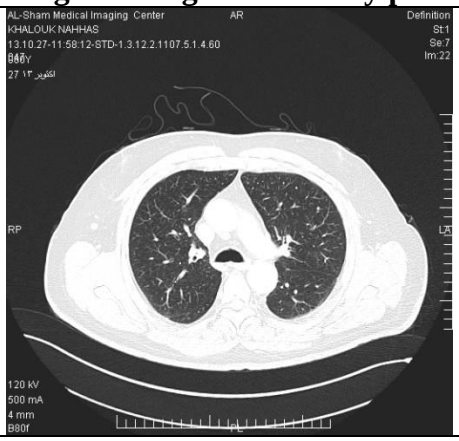
A lung CT image of a healthy person	A lung CT image of a cancer patient
	
534x512	703x512

Fig. 1. A CT image of a healthy person (left), and another one of a lung cancer patient (right). The nodules diagnosed by doctor, are surrounded by a small circles.

4. NODULES AUTOMATIC EXTRACTION METHOD

As can be seen in Fig. 1, the original lungs CT image is a complicated content one because it contains, as well as the lungs area, the name of the medical center, the name of the patient, the date, and several other types of information and shapes to help the doctor in diagnosis and archiving. Besides, the lung nodes and the nodules are rather similar in shape, and their gray levels are similar to those of the surrounding region. This situation makes direct extraction of the lungs with their pictorial content from the original image (by thresholding, for example) impossible. Therefore, the general method we use to extract the nodules consists of two main stages, as shown in Fig. 2. Of course, each stage consists of several steps. In the first stage, the lungs area is extracted from the original CT image, and in the second one, the nodules in the lungs area are detected and extracted. If no nodules found in the lungs image, it is marked as "healthy lungs". The output image contains the extracted nodules that can be used later in diagnosis research.

4.1. Lungs Area Extraction Stage

The automatic extraction of lungs area from the abdominal CT image is a complicated process. It is rich of details that we should consider in order to extract the lungs accurately from different CT shapes. Therefore, we explain the steps of this stage rather briefly here, then explain later the nodules detection and extraction process. In this stage, we extract the lungs area from the original chest image through two essential processes. In the first one, the known "lungs mask" is extracted through several steps. Then the mask is multiplied by the original image to extract the lungs area from the chest image with the original gray levels preserved. We explain the complete steps in the following:

1– Thresholding the input original image automatically using Outsu's method [15] to select the threshold, then all pixels values in the image below this threshold are set to zero "0", and the rest ones are set to "1", resulting in a binary image. Fig. 3(b) shows the result of thresholding the original image shown in Fig. 3(a).

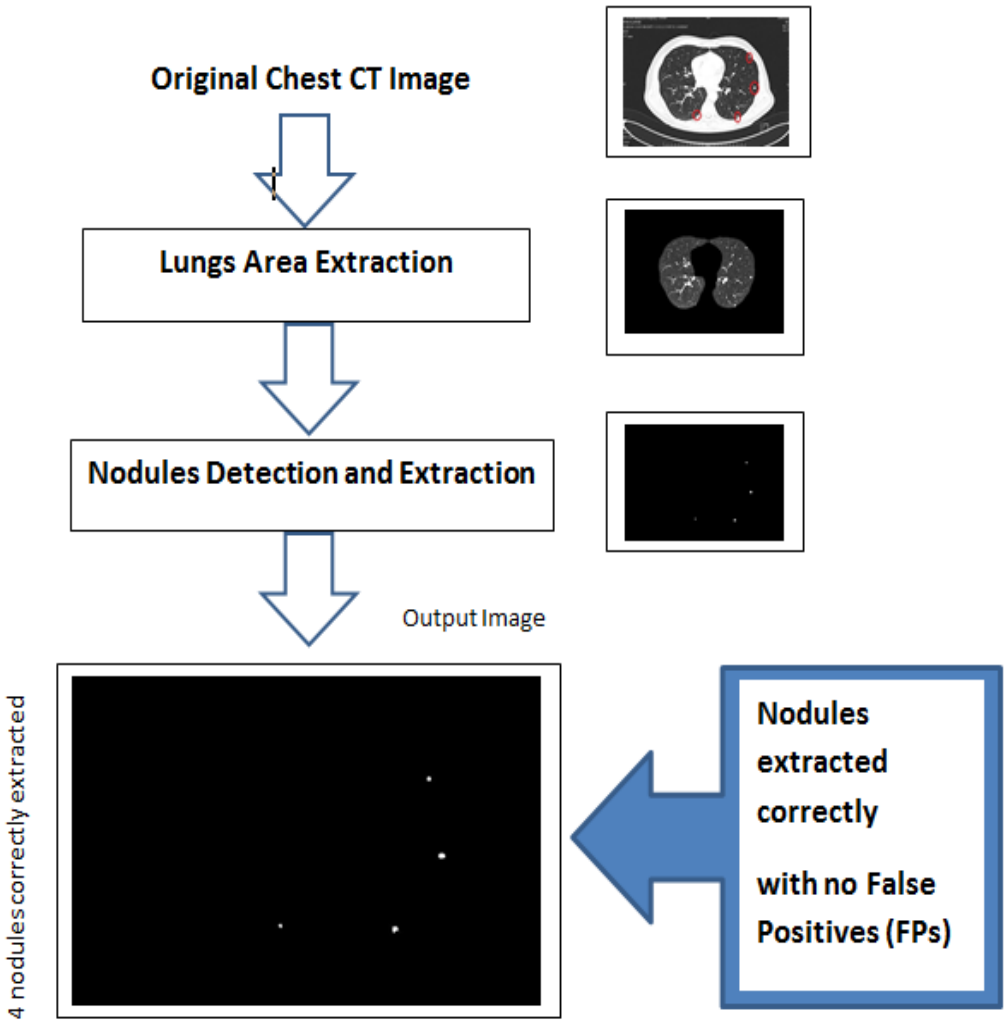


Fig. 2. The two main stages of the general nodules extraction method with results.

2- Labeling the connected components in the thresholded image to give each distinct object (connected component) a unique label that enables us, in principle, to compute all possible kinds of features and use them appropriately to extract lungs area or to distinguish the nodules.

3– Finding the largest component which is the closed region surrounding the lungs, by selecting the component that has the maximum area measured by number of pixels in the thresholded image, Fig. 3(c) shows the largest component obtained from Fig. 3(b).

4– Applying a "closing" process to the complement of Fig. 3(c) to remove any remaining printed characters or tiny objects attached to the largest component and to close small holes. The result of this process is shown in Fig. 3 (d).

5 – Applying a hole-filling process to the complement of Fig. 3 (d) to get the complete inner area of the largest component, shown in Fig. 3 (e).

6- Removing any organs may remain near the lung: This is done by applying an "opening" process followed by a "closing" one using a circular "structuring element" with a diameter (D=10). This case does not appear in some CT scans. Fig.3 (g) shows an organ removed from Fig. 3(f) by this step. Note that we use here a different image to show this case, which does not appear in all CT images.

7- Multiplying Fig. 3(d) by Fig. 3(e) to get what we called Lung Mask shown in Fig. 3 (h).

8- Finally, multiplying the Lung Mask by the original image in Fig. 3(a) to extract the lungs area with original gray levels preserved, as shown in Fig. 3 (i).

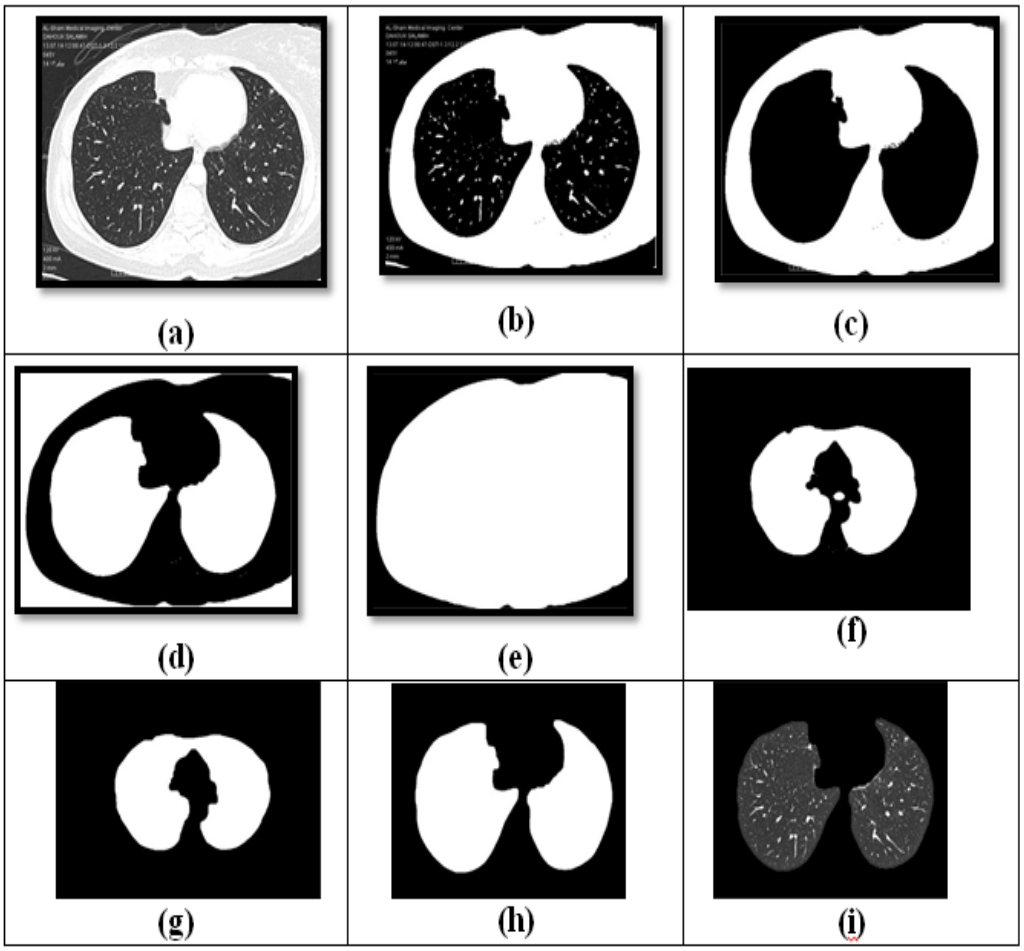


Fig. 3. Sample results of the eight steps lungs extraction process.

The method explained above extracted successfully lungs areas from CT images of different general shapes and complications, as shown in Fig. 4. These results give an idea of the flexibility and effectiveness of the proposed method.

4.2. Extraction of nodules from lungs image

After having the lungs area extracted, we extract nodules through three main stages: (1) Preparing the lungs area image for feature extraction, (2) Extracting features used for classification, and (3) Classifying the regions remaining in the prepared image into "nodules", and "not nodules". Regions classified as "nodules" remain in the resultant image and the others are removed, as shown in Fig. 2. We explain the three stages in the following in necessary details.

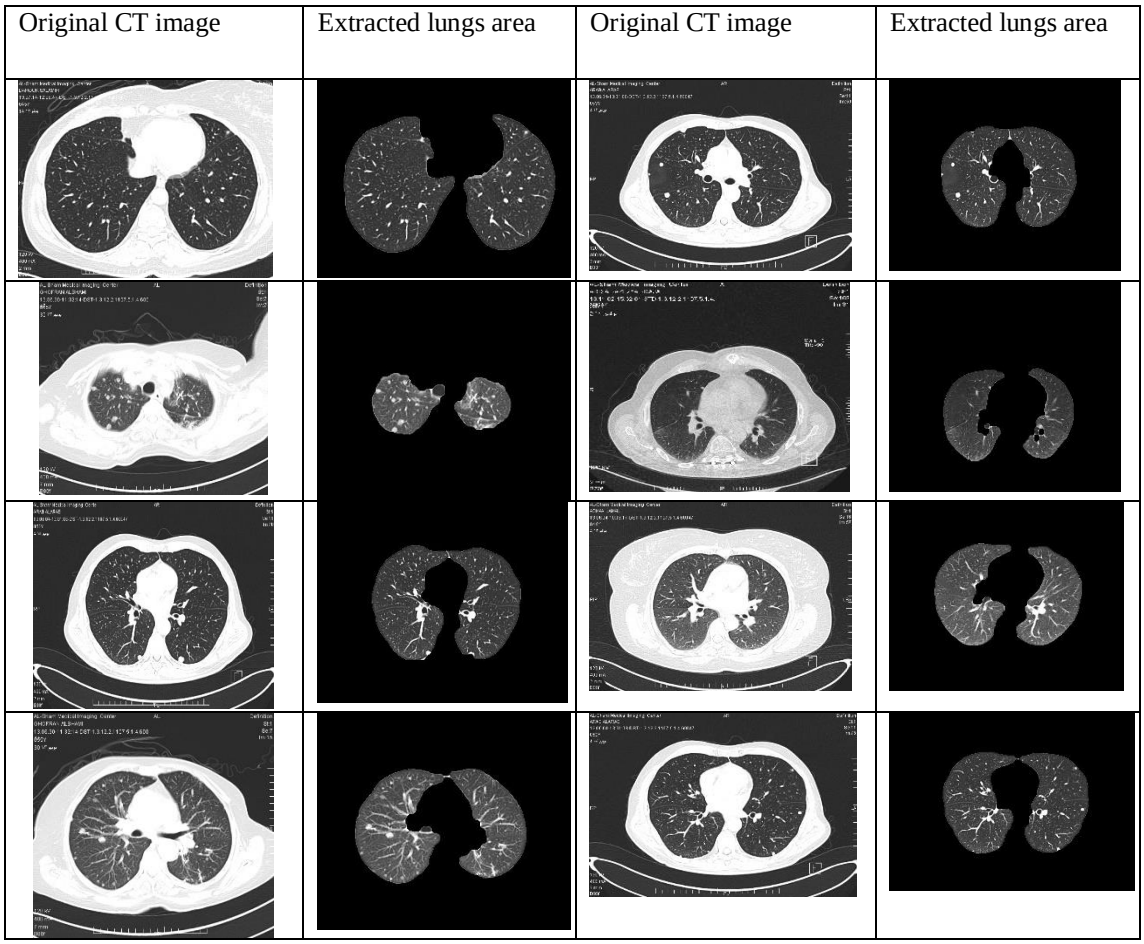
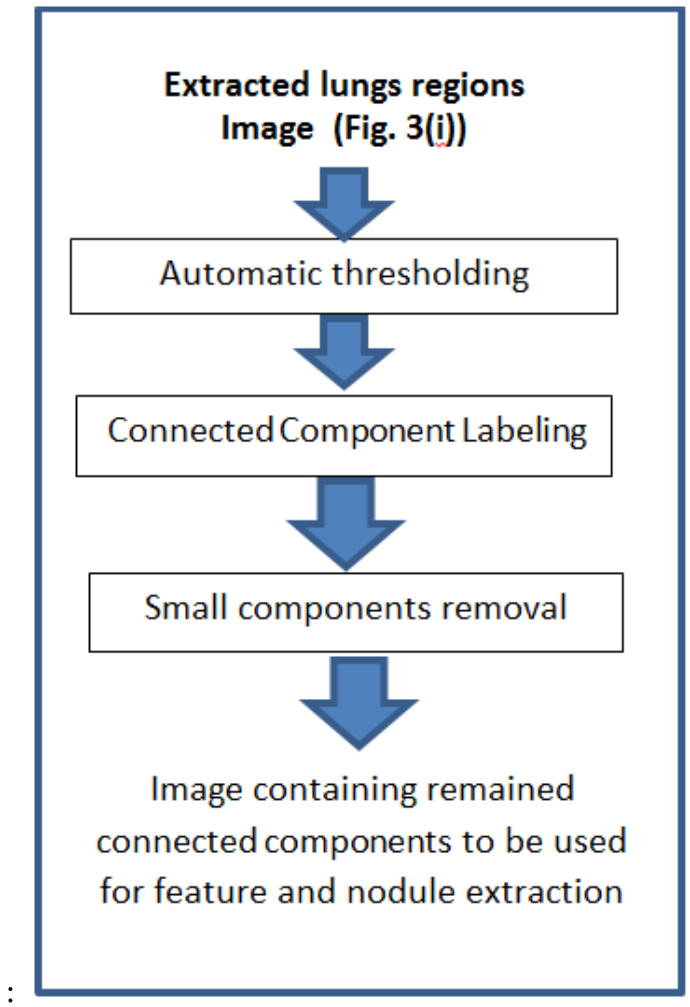


Fig. 4. Sample results of extracting lungs areas from images of different general shapes and complication, with very good accuracy.

4.2.1. Preparing the lung area for feature extraction

Preparing the image of lungs area for feature extraction is done in three steps, as shown in Fig. 5. These steps are:

(1) Binarizing the image using Outs's method to select the threshold in the same way used in step 1 of lungs area extraction process explained in section 4-1.
(2) Labeling connected components in the lungs area binary image.
(3) Removing small components with areas less than 15 pixels, since as our investigation of all images showed that the regions of such areas are not nodules. Removing these small components will save some of the computation time needed for feature extraction and classification. Fig. 6 shows a sample result with the remaining CCs.



:

Fig. 5. Preparing the image used for feature extraction.

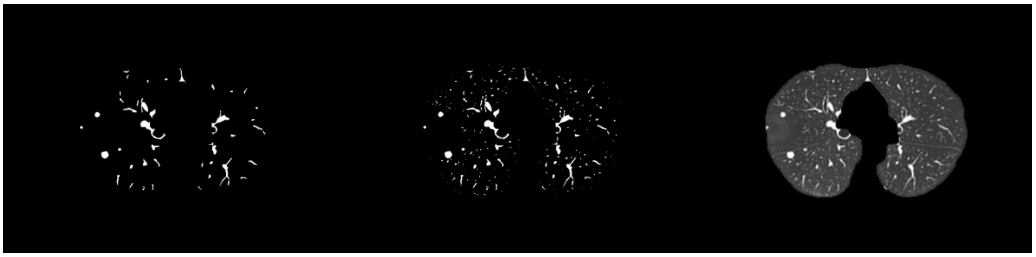


Fig. 6. An extracted lungs image (left), binarized image (middle), remaining CCs after small components removing (right).

4.2.2. Feature extraction

Specialized doctors diagnosed the nodules used in this study. We found by visual interactive investigation of our data using MATLAP programming facilities that, in general, the nodules have a rather circular shape with almost specific distribution of gray levels, Fig. 7 shows two examples of the interactive examination we made. Therefore, we extracted two groups of features: shape features group and density features group. We will explain the two groups in the following two subsections.

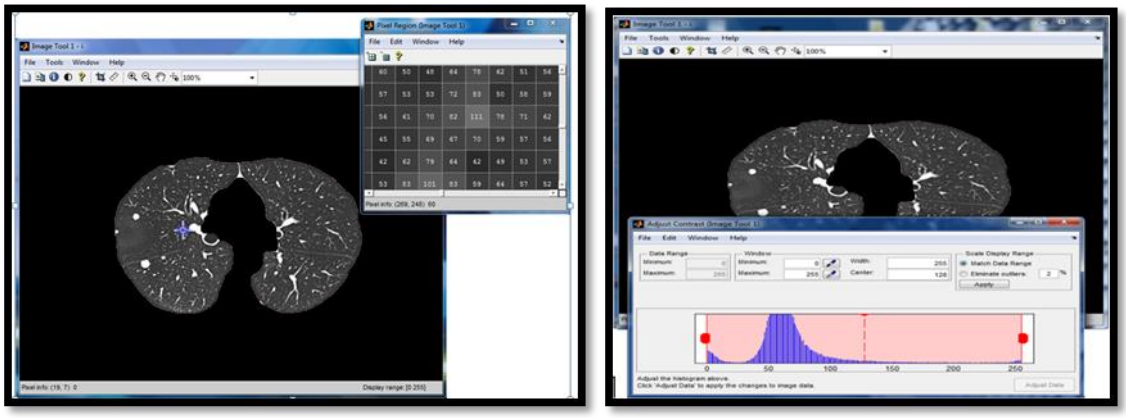


Fig. 7. Two examples (gray level measuring and threshold selection) of the response of the visual interactive image examination using MATLAP facilities we used.

4.2.2.1. Shape Features used in the Study

After labeling the remaining components in the lung image, we extracted the following shape features for every component:

$\mathbf{F_s1:}\ form\ factor = \frac{4\pi * Area}{Perimeter}$; $\mathbf{Fs2:}\ roundness = \frac{4 * Area}{\pi * MaxDiameter^2}$

$\mathbf{F_s3:}\ \text{solidity} = \frac{Area}{ConvexArea}$; $\mathbf{F_s4:}\ \text{extent} = \frac{TotalArea}{Area\ Bounding\ Rectangle}$

$\mathbf{F_s5:}\ \text{compactness} = \frac{\sqrt{(4 * Area)/\pi}}{MaxDiameter}$; $\mathbf{F_s6:}\ \text{aspect ratio} = \frac{MaxDiameter}{MinDiameter}$

Where F_s stands for: shape feature.

4.2.2.2. Density features

Density features are those related to gray levels of image components. Several density features can be extracted from a gray image [16]. We extracted the following ones:

F_d1: 3rd normalized moment: $\mu_2 = 255 * \sum_{i=0}^{255} (z_i - m)^2 p(z_i)$

F_d2: 4th normalized moment: $\mu_3 = 255 * \sum_{i=0}^{255} (z_i - m)^3 p(z_i)$

F_d3: standard deviation: $\sigma = \sqrt{\mu_2}$

F_d4: Smoothness measure: $R = 1 - \frac{1}{1-\sigma^2}$

F_d5: similarity measure: $U = \sum_{i=0}^{255} p^2(z_i)$

F_d6: entropy: $e = -\sum_{i=0}^{255} p(z_i) * log_2\{p(z_i)\}$

Where: $m = \sum_{i=0}^{255} z_i * p(z_i)$, F_d: density feature, m: average of gray levels; z_i: value of the pixel at gray level (i); $p(z_i)$:value of the histogram at gray level (i).

4.2.3. Detection of nodules from extracted lungs with remaining CCs

The image of the extracted lungs area with the remaining connected components CCs, like that in Fig. 6, contains nodules and none nodules CCs. Therefore, detecting nodules in this situation

belongs to the Pattern Recognition standard two-classes problem, in which a test sample must be classified whether to belong to the first class (nodule, here), or to the second class (not-nodule). This situation is exactly similar to signature verification problem in which the test signature must be judged whether to be genuine or a forgery. Ammar M. developed a reliable system for signature verification that gives its decision based on a threshold on a Weighted Euclidian Distance Measure computed from suitable features [17]. This system is US patented [18], and still working commercially in hundreds of US banks since about two decades examining about 3 million checks every day. Based on that, this approach is expected to work well in nodule detection. We detected nodules-components by computing the Weighted Euclidean Distance (WED) of each CC in the image using shape and density features explained in the previous two subsections, then using a global threshold obtained from the training group, we classified each CC with a WED less than this global threshold as a "nodule", because the low WED means that the CC resembles the nodules more than the arbitrary CCs. CCs that have a WED equal or larger than the threshold is considered as not-nodule.

5. TRAINING

The training to obtain the global threshold is done on 30 images among the 98 ones containing nodules diagnosed by the specialized doctor. Fig. 8, for example, shows the feature values and the WED of the 4 CCs in a simple case image containing one nodule diagnosed by the doctor, and denoted by "T" in the last column. It is obvious that this nodule is easily separable from the other CCs due to the considerable difference in the WED value. This was an example, but the complete training was done using all shape and density features in 30 images from the 98 images containing nodules used in this study. It is worth noting that the number of remaining CCs in an image reaches 41 in some images.

CC #	form factor	roundness	Solidity	Extent	Compactness	Aspect ratio	Distance	Doc. Diag.
1	12.0391719	1001.60565	0.833333	0.535714	31.64815402	2.188309697	1.539006	
2	16.5141083	432.901445	0.93	0.9375	20.80628379	1.136570857	1.435795	
3	31.7389353	23175.2454	0.886957	0.653846	152.2341793	1.281090021	0.201863	T
4	17.431828	7495.76987	0.82	0.414141	86.57811425	2.48330078	1.306872	

Fig. 8. Shape feature values and Weighted Euclidean Distance Measure of the remaining CCs of a simple test image.

6. RESULTS AND DISCUSSION

After training on the 30 images using shape features once, and both shape and density features once again to obtain the global threshold that separates the nodules from the other CCs in each case, we obtained the results shown in Table 1, where: TP is "True Positive", "FP" is "False Positive", "FN" is False Negative, and the "Sensitivity" is computed by this equation: Sensitivity = TP/(TP+FN).

Table 1. Results of the nodules detection using shape and density features on test cases of 68 images.

Feature kind	Sensitivity%	FPs (per case)	No. CCs	No. of Nodules Diagnosed by doctor
Shape features	95.1	2.1	1836	173
Shape +Density features	97.2	1.98	1836	173

We can conclude from Table 1 that: the average diagnosed nodules is about 2.54 per image, and the average number of remaining CCs in an image before detection and extraction is 27. It is worth noting that FPs is a false alarm, and it is not dangerous.

Using the density features with shape features improves the sensitivity by about 2%, and the improvement in FPs reaches about 6%.

6.1. Performance consistency check

As a consistency check of the performance of the nodule detection and extraction algorithm, we tested the lung images of the 15 healthy persons as a consistency-test-group. The result was excellent where 9 false positives (FPs) appeared in 8 images (4% of CCs, as an average). Fig. 9 shows the result for a test image containing seven nodules where all detected with one FP, and the second is for a healthy person where no nodules are detected, and consequently will be classified as "healthy lung" .

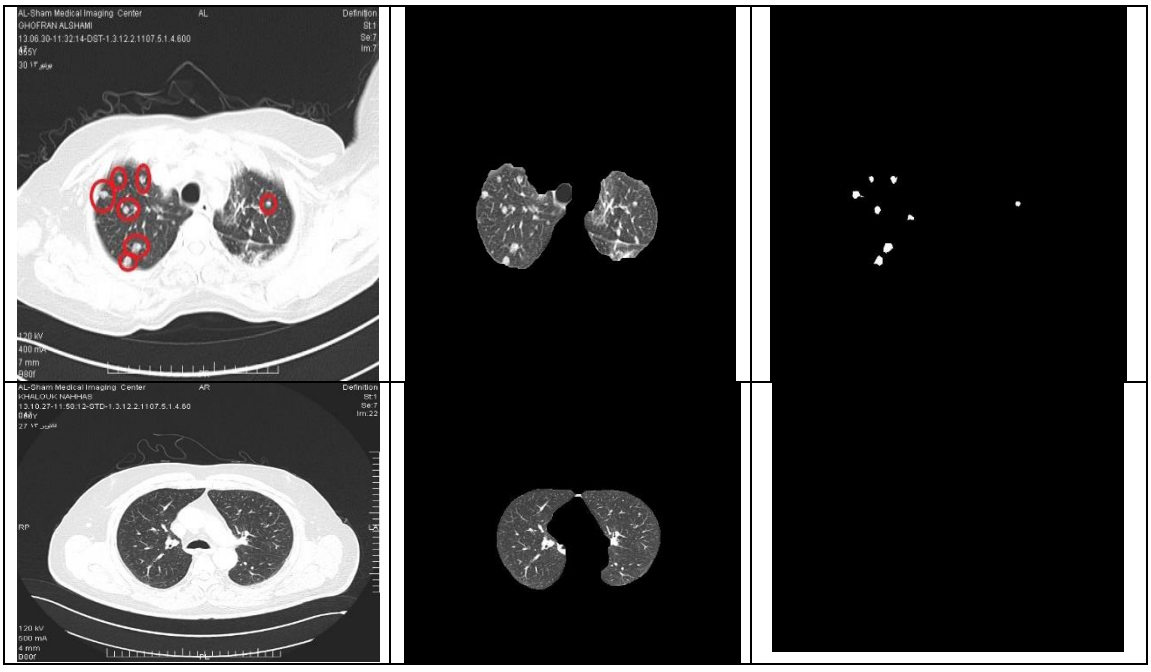


Fig. 9. The first column is the input image, the second column is the extracted lungs image, and the third column is the result of detection and extraction of nodules. In the first row: TPs=7, and FPs=1. In the 2nd row: TPs=0, FPs=0, healthy person (perfect result).

6.2. Comparison with some other works

As can be seen in Table 2 below, the introduced method gave a general performance compared with the best results reported, with rather higher sensitivity and it approached the minimum (FPs) false positives.

Table 2. Performance comparison between other works and the introduced method.

Authors	year	Sensitivity%	FPs per case	No. nodules
Liu [19]	2010	97	4.3	32
Cascio [20]	2012	97	6.1	148
Orozco [21]	2012	96.15	2	50
Teramoto [22]	2013	80	4.2	103
Shao [23]	2012	89.47	11.9	44
Bergtholdt [24]	2016	85.90	-	
Wu [25]	2017	79.23	-	
Saien [26]	2018	83.98	0.02	
Khehrah [12]	2020	93.75	0.13	
Monif(our work)	2021	97.2	2.1	173

7. FUTURE WORK

We plan to work on improving the performance of the classification by using the feature selection technique developed by Ammar M. [17].

8. CONCLUSIONS

We have introduced a method for the automatic detection and extraction of lungs cancer nodules from CT images using connected component labeling (CCL) and weighted distance measure based classification. The obtained results have shown clearly the high performance of the method in both extraction of lungs area from the CT image and in the correct detection of the cancer nodules. This high performance was also supported by the consistency check we made which reveals the ability of the method to detect the healthy image at the same time.

The experiments conducted and their results analysis presented in this paper enable us to conclude that using CCL technique with appropriate sequence and parameters of some morphological operations may lead to a high performance approach for extracting lungs areas from complex chest CT image with a wide variety in shapes. We can conclude also that using CCL technique with the high flexibility it offers in manipulating CCs in the extracted lungs areas, with the usage of WEDM and a threshold based classification, may provide an efficient method for detecting and extracting nodules to be used later for diagnosis. We found also that using density features with shape ones improves to some extent the performance. We hope that we have introduced a positive effort in the general direction of the research for building an actual automatic lung cancer detection and diagnosis systems.

ACKNOWLEDGMENTS

The authors wish to thank Alsham Medical Imaging Center and Tishreen Hospital for providing the lung images data. We appreciate also the support provided by Al Andalus University for Medical Sciences and its hospital.

REFERENCES

[1] WHO, Latest world cancer statistics, (2013), The International Agency for Research on Cancer Publications, Geneva: World Health Organization.

[2] Ayman El-Baz, et al.,(2013) Computer-Aided Diagnosis Systems for Lung Cancer: Challenges and Methodologies, International Journal of Biomedical Imaging, Volume 2013, Article ID 942353.

[3] Heang-Ping Chan et al, (2008) Computer-Aided Diagnosis of Lung Cancer and Pulmonary Embolism in Computed Tomography — A Review, Acad. Radiology, 15(5): 535–555.

[4] Sprindzuk M.V., et al., (2010) Lung cancer differential diagnosis based on the computer assisted radiology: The state of the art, Pol J Radiology, 75(1): 67–80.

[5] Firmino et al., (2014) Computer-aided detection system for lung cancer in computed tomography scans: Review and future prospects, BioMedical Engineering, 13:41.

[6] Bhavanishankar K. et al., (2015) techniques for detection of solitary pulmonary nodules in human lung. and their classifications-A survey, International Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 1.

[7] Kaur R., and Ada S. (2013), "Feature Extraction and Principal Component Analysis for Lung Cancer Detection in CT scan Images", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 3, pp.187-190.

[8] Miwa T., Kako J., Yamamoto S., Matsumoto M., Tateno Y., Iinuma T., and Matsumoto T. (2002), "Automatic Detection of Lung Cancers in Chest CT Images by the Variable N-Quoit Filter", Systems and Computers in Japan, Vol. 33, No. 1, pp.53-63.

[9] Homma N., Takei K. and Ishibashi T. (2008), "Combinatorial Effect of Various Features Extraction on Computer Aided Detection of Pulmonary Nodules in X-ray CT Images", INFORMATION SCIENCE & APPLICATIONS, Issue 7, Vol. 5, pp.1127-1136.

[10] Gomathi M., and Thangaraj P. (2010), "A Computer Aided Diagnosis System For Detection Of Lung Cancer Nodules Using Extreme Learning Machine", International Journal of Engineering Science and Technology, Vol. 2, pp. 5770-5779.

[11] Wang S., Dong L., Wang X., Xin Wang, Open Med. (2020) Classification of pathological types of lung cancer from CT images by deep residual neural networks with transfer learning strategy.

[12] Khehrah N., Farid M. S., Bilal S. and Khan M. H., (2020) Lung Nodule Detection in CT Images Using Statistical and Shape-Based Features, J. Imaging, 6, 6 (14 pages).

[13] MakajuS., Prasad P.W.C., Alsadoon A., Singh A. K. and Elchouemi A., (2018) Lung Cancer Detection using CT Scan Images, Procedia Computer Science 125- 107–114.

[14] Ammar M., et al., (2011) Using Image Processing Techniques for Automatic Extraction of Liver Suspicious Regions from X-Ray Computed Tomography Images, Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series, Vol. (33) No. (3), pp.(217-235).

[15] *Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9(1): 62–66*.

[16] Gonzalez, R., Wood, E. and Eddins, S. (2009) "Digital Image Processing Using MATLAB", Second Edition, Gatesmark Publishing.

[17] Ammar M., Raising the Performance of Automatic Signature Verification Over that Obtainable by Using the Best Feature Set, (2011) International Journal of Pattern Recognition and Artificial Intelligence. 25-2, PP 183-206.

[18] Ammar M., Method and apparatus for verification of signatures, United States Patent**:** No. 6424728 , 07/23/2002, U.S.A.

[19] Liu Y, Yang J, Zhao D, Liu J (2010) A method of pulmonary nodule detection utilizing multiple support vector machines. In Computer Application and System Modeling (ICCASM), International Conference On, vol. 10. Taiyuan; 10–11810121.

[20] Cascio D, Magro R, Fauci F, Iacomi M, Raso G (2012)Automatic detection of lung nodules in CT datasets based on stable 3d mass-spring models. ComputBiol Med, 42(11):1098–1109.

[21] Orozco HM, Osiris Vergara Villegas O, Maynez LO, Sanchez VGC, de Jesus Ochoa Dominguez H (2012) Lung nodule classification in frequency domain using support vector machines. In Information Science, Signal Processing and Their Applications (ISSPA), 11th International Conference On. Montreal, QC; 870–875.

[22] Teramoto A, Fujita H (2013) Fast lung nodule detection in chest CT images using cylindrical nodule-enhancement filter. Int J Comput Assist RadiolSurg, 8(2):193–205.

[23] Shao H, Cao L, Liu Y (2012) A detection approach for solitary pulmonary nodules based on CT images. In Computer Science and Network Technology (ICCSNT), 2nd International Conference On. Changchun; 1253–1257.
[24] Bergtholdt, M.; Wiemker, R.; Klinder, T. (2016) Pulmonary nodule detection using a cascaded SVM classifier. InProceedings of the Medical Imaging: Computer-Aided Diagnosis, San Diego, CA, USA, 27 February–3March 2016; Volume 9785, pp. 268–278.
[25] Wu, P.; Xia, K.; Yu, H. (2016) Correlation coefficient based supervised locally linear embedding for pulmonary nodule recognition. Comput. Methods Programs Biomed. 136, 97–106.
[26] Saien, S.; Moghaddam, H.A.; Fathian, M. (2018) A unified methodology based on sparse field level sets and boosting algorithms for false positives reduction in lung nodules detection. Int. J. Comput. Assist. Radiol. Surg., 13, 397–409.

AUTHORS

Mamdouh Monif PhD in Biomedical Engineering from Technical University of Graz, Austria, 2000, Associate Professor and Dean of the Faculty of Biomedical Engineering - Al-Andalus University for Medical Sciences, 2020 to present. US patent: SYSTEMS AND METHODS TO MEASURE FLUID IN A BODY SEGMENT، US 9895069 B2; Feb. 2018. Published several papers in biomedical engineering field.

Kenan Mansour MD Obstetrics and Gynaecology Specialist, Tishreen Hospital, 2011. Medical Manager of Al Andalus University Hospital, 2019 to present. Medical Education Master student, Syrian Virtual University, 2020. Interested in medical image analysis and diagnosis.

Waad Ammar MD General Surgery Specialist, Tishreen Hospital, 2020. At present, working at Al Andalus University Hospital. Medical Education Master student, Syrian Virtual University, 2020. Interested in medical image analysis and diagnosis.

Maan Ammar Ph. D. in Information Engineering, Nagoya University, Japan, 1989, Professor at Al Andalus University for medical sciences , Biomedical engineering since 2014, Full professor at Applied Sciences University, Amman Jordan 2003, US patent of a commercial system serving hundreds of US banks since 2002 "Method and apparatus for verification of signatures", United States Patent: No. 6424728 , 07/23/2002, U.S.A. Published many papers in image processing and pattern recognition fields. Served as Head of biomedical engineering department–Damascus University for 8 years.

THE CASE FOR ERROR-BOUNDED LOSSY FLOATING-POINT DATA COMPRESSION ON INTERCONNECTION NETWORKS

Yao Hu and Michihiro Koibuchi

National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan

ABSTRACT

Data compression virtually increases the effective network bandwidth on an interconnection network of parallel computers. Although a floating-point dataset is frequently exchanged between compute nodes in parallel applications, its compression ratio often becomes low when using simple lossless compression algorithms. In this study, we aggressively introduce a lossy compression algorithm for floating-point values on interconnection networks. We take an application-level compression for providing high portability: a source process compresses communication datasets at an MPI parallel program, and a destination process decompresses them. Since recent interconnection networks are latency-sensitive, sophisticated lossy compression techniques that introduce large compression overhead are not suitable for compressing communication data. In this context, we apply a linear predictor with the user-defined error bound to the compression of communication datasets. We design, implement, and evaluate the compression technique for the floating-point communication datasets generated in MPI parallel programs, i.e., Ping Pong, Himeno, K-means Clustering, and Fast Fourier Transform (FFT). The proposed compression technique achieves 2.4x, 6.6x, 4.3x and 2.7x compression ratio for Ping Pong, Himeno, K-means and FFT at the cost of the moderate decrease of quality of results (error bound is 10^{-4}), thus achieving 2.1x, 1.7x, 2.0x and 2.4x speedup of the execution time, respectively. More generally, our cycle-accurate network simulation shows that a high compression ratio provides comparably low communication latency, and significantly improves effective network throughput on typical synthetic traffic patterns when compared to no data compression on a conventional interconnection network.

KEYWORDS

Interconnection Network, Lossy Compression, Floating-point Number, Linear Predictor, High-performance Computing (HPC).

1. INTRODUCTION

The network bandwidth becomes one of the primary concerns on recent parallel computers, because its annual improvement is moderate compared to that of computation power in compute nodes. A way to virtually increase the network bandwidth is the reduction of redundancy of communication data themselves. Some parallel scientific applications repeatedly generate similar communication data [1]. For this kind of communication, each compute node has a chance to reduce traffic by compressing communication data, e.g., by sending only the information of difference from the prior data.

A floating-point dataset is frequently exchanged between compute nodes in parallel applications. However, its compression ratio generally becomes low when using a simple lossless compression algorithm. A complicated lossless compression algorithm would have a high compression ratio.

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 55-76, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110706

However, it has a large latency overhead to compress communication data. It is not suitable for interconnection networks, because HPC interconnection networks are latency-sensitive, e.g., less than one microsecond for inter-process communication [2].

In this study, we aggressively apply a fast lossy compression algorithm to IEEE 754 floating-point communication data on interconnection networks. Generally, a lossy compression achieves a higher compression ratio than that by a counterpart lossless compression for a given dataset.

We take an application-level compression for providing high portability. In parallel programs, floating-point values are compressed before they are sent at the source side, and exchanged to relevant processes. They are then decompressed at the receiver side. The compression algorithm relies on value predictions for floating-point values, like SZ [3]. We provide byte- and bit-wise compression techniques to MPI (message passing interface) implementation. If the value prediction succeeds in the byte-wise compression, the floating-point value is converted to a single byte expression, corresponding to an MPI char type. Otherwise, the original value is not compressed, and it is transferred as it is. Although the byte-wise compression has low compression-operation latency, its upper bound of the compression ratio is not high, i.e., obviously four and eight for single-precision and double-precision floating-point values, respectively. By contrast, the bit-wise compression generates a bitstream encapsulated in a byte array, corresponding to an MPI char type as well. If the value prediction succeeds at a source, the floating-point value is converted to three bits. Even if the value prediction fails, the least significant bits (LSBs) are discarded from the IEEE 754 floating-point expression of the value to obtain a relatively high compression ratio, while maintaining a given error bound.

In this study, the loss during data compression is restricted within a specified absolute error bound to have acceptable execution results for target MPI applications. Both approaches work well in cases where subsequent data correlate reasonably with earlier data, which is often the case for the intermediate and final floating-point results produced by some scientific programs. In this case, approximate communication can be applied to error-resilient applications for speedup while maintaining the accuracy of the output at an acceptable level. In our study, several representative MPI-based error-resilient applications are used for the evaluation of the proposed lossy compression algorithm. Their computing tasks either do not aim at an exact numerical answer or they have inherent resilience to output error.

Our main contributions in this work are as follows:

- We designed, implemented, and evaluated byte- and bit-wise lossy application-level compression techniques based on several predication models for floating-point communication values in MPI parallel applications.
- We obtained 2.4x, 6.6x, 4.3x and 2.7x compression ratio (error bound is 10^{-4}) for Ping Pong, Himeno, K-means clustering and FFT applications, thus speeding up the execution time by 2.1x, 1.7x, 2.0x and 2.4x, respectively.
- From the network point of view, our cycle-accurate network simulation shows that, when the compression ratios become 1.5, 3.0 and 6.0, the effective network throughput is improved by 133%, 176% and 260%, respectively.

The rest of this work is organized as follows. Background information and related work are discussed in Section 2. Section 3 presents the byte- and bit-wise compression techniques. Section 4 shows evaluation methodology and results. Section 5 concludes with a summary of our findings in this work.

2. BACK GROUND AND RELATED WORK

2.1. Lossy Compression

Lossy compression or irreversible compression is a class of data encoding methods, which uses inexact approximation and partial data discarding to represent the content. Lossy compression is opposed to lossless compression which does not degrade the data precision or quality. The compression ratio of lossy compression is usually higher than that of lossless compression due to the data precision loss to a limited extent.

A widely used lossy compression algorithm is the discrete cosine transform (DCT) [4], which is most commonly used to compress multimedia data such as audio, video and images. Another famous lossy compression algorithm of the transform type is fast Fourier transform (FFT) [5], which converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.

Lossy compression of the predictive type was applied to differential pulse-code modulation (DPCM) [6], which is used as a signal encoder that uses the baseline of pulse-code modulation (PCM) but adds some functionalities based on the prediction of the samples of the signal. Another proposed lossy compression algorithm called linear predictive coding (LPC) [7] is widely used in speech coding and speech synthesis, using the information of a linear predictive model to represent the spectral envelope of a digital signal of speech in compressed form in audio signal processing and speech processing. FPZIP [8] was primarily designed for lossless compression but also has provision for lossy compression to achieve a high compression ratio. It predicts the data by a subset of encoded data and maps the difference to an integer number. For lossy compression, ZFP [9] often outperforms FPZIP by dividing 3-D floating-point arrays into small and fixed-size blocks of dimensions to achieve a higher compression ratio. A more complicated lossy compression algorithm called ISABELA [10] can achieve a higher compression ratio by transforming data layout such as sorting, cubic B-splines fitting and window splitting.

The prediction techniques based on the preceding data elements have received much attention for both lossless compression and lossy compression in recent years. For instance, the works [11] [12] [13] extend predictive coding to floating-point data and compress floating-point values without loss. The predicted and the actual floating-point values are broken up into sign, exponent and mantissa, and their corrections are compressed separately with context-based arithmetic coding. FPC [14] [15] [16] is a high-speed compressor for double-precision floating-point data. In FPC, the data are predicted by using FCM/DFCM (differential-finite-context-method predictor) and selecting closer values to the true ones. The bit-wise XOR operations are then performed between the predicted values and true values. Finally, the leading zeros in the result are compressed to less (e.g., 4) bits. SZ [3] [17] [18] is proposed for lossy compression by predicting data using three curve-fitting models, i.e., preceding-neighbor fitting model, linear-curve fitting model and quadratic-curve fitting model. The essential idea of SZ is to use linear predictive coding for predictable data and to perform complicated binary analysis for unpredictable data. Another work [1] presented a similar idea of floating-point data compression for FPGA-based high-performance computing by using a one-dimensional polynomial predictor.
The above lossy compression algorithms usually have a tradeoff between the compression latency overhead and the compression ratio. Most of them are historically optimized for the purpose of storing compact data in storage. Even for recent HPC and cloud purposes, the main target is still to compress data on storage [19], e.g., storing checkpoint images at the cost of trivial decrease on quality of results [20]. Our target, the interconnection network, is radically different

from that assumed in most of existing compression algorithms. Compressing data for inter-process communication in parallel programs is latency-sensitive when compared to that for storage. A recent work [34] proposed a DCT-based approximate communication scheme to reduce communication overhead without a considerable quality loss of the result. It obtains a good balance between compression speed and compression ratio. However, its limitation is that it is only useful for non-random message patterns. To the best of our knowledge, this study is the first challenge to apply a prediction-based lossy data compression algorithm at a program level to the inter-process communication datasets generated in MPI parallel applications.

2.2. Data Compression on Interconnection Networks

Besides off-chip interconnection networks, there are some prior works on data compression techniques targeting interconnection networks on a chip. Lossless frequent-pattern compression (FPC) is a significance-based compression scheme for on-chip communication to cache [21]. It compresses only some data patterns, such as leading 0s and 1s in data streams on the network-interface hardware. FPC introduces a variable-length cache line, and its essential design is to handle variable cache line sizes [22]. It enables low compression overhead, e.g., one or two clock cycles. FPC is efficient for fixed-point values, e.g., integer numbers. However, it does not work well for IEEE 754 floating-point values. This is because a floating-point value includes many mantissa bits that hardly include locality. The work [23] on lossy compression of floating-point data provides a fast lossy compression scheme which simply truncates the 16, 24 or 32 least-significant bits to save total link energy. However, the error bound is not guaranteed during compression especially for near-zero small floating-point values, which is distinct from our error-bounded lossy compression techniques.

3. ERROR-BOUND LOSSY FLOATING-POINT COMPRESSION TECHNIQUES

In this section, we present byte- and bit-wise prediction-based compression techniques that have different tradeoffs between the compression latency overhead and the compression ratio.

3.1. Linear-predictive Byte-wise Compression

A compression algorithm often encodes the differences or XORs between the given input data and the one predicted by using previous data. In this study, we map the input data to predefined symbols if the given data can be predicted by their previous data within the error bound.

3.1.1. Design

Based on how prediction is actually performed, prediction-based algorithms are classified into two groups: arithmetic-based algorithms [11] [12] [13] [8] and context-based algorithms [14] [15] [16]. Arithmetic-based algorithms use an arithmetic predictor to obtain prediction via calculations, whereas context-based algorithms use hash tables to look up data that appear after the same input phrase to predict the next input. We choose the arithmetic-based algorithmic approach because our target is IEEE 754 (single- and double-precision) floating-point formats with almost no bit-level locality for expressing two similar values. Although the arithmetic-based prediction requires buffer memory to retain previous inputs, the size of the buffer can be reduced by employing one-dimensional polynomial predictions [1].

We use a versatile linear predictor for lossy compression [24] [3] [1]. For a one-dimensional numerical dataset $d = \{d_1, d_2, \ldots, d_M\}$, the linear predictor predicts each element d_i by its $n + 1$ preceding elements $\{d_{i-(n+1)}, \ldots, d_{i-1}\}$ within the given error bound. We use typical polynomial

predictions, p_i, according to varying n. The predictions for the first four values of n are as follows:

$$p_i^0 = d_{i-1} \ (n = 0) \quad (1)$$

$$p_i^1 = 2 \times d_{i-1} - d_{i-2} \ (n = 1) \quad (2)$$

$$p_i^2 = 3 \times d_{i-1} - 3 \times d_{i-2} + d_{i-3} \ (n = 2) \quad (3)$$

$$p_i^3 = 4 \times d_{i-1} - 6 \times d_{i-2} + 4 \times d_{i-3} - d_{i-4} \ (n = 3) \quad (4)$$

The byte-wise compression is illustrated in Algorithm 1. The basic idea is borrowed from the bestfit curve-fitting algorithm in [3]. In Line 3, the conversion is performed to handle bit strings of floating-point values such that each element of d is not less than zero, i.e., $d_i \geq 0$. This conversion ensures that the sign bits of negative floating-point values are flipped to zero, which improves the performance of the linear prediction.

Algorithm 1 The byte-wise compression.

Input:
1-D floating-point array D, and user-defined error bound E
Output:
1-D byte array D_{cmp}, 1-D floating-point array D_{uncmp}, and 1-D integer array D_{index}

```
1:  for i = 1 → M do
2:      /* Difference Preprocessing */
3:      convert D_i into non-negative floating-point d_i
4:      bestfit = arg min (|p_i^0 − d_i|, |p_i^1 − d_i|, |p_i^2 − d_i|, |p_i^3 − d_i|)
5:      if p_i^bestfit ≤ E then
6:          switch bestfit do
7:              case 0: append (00)_16 to D_cmp /* 1 byte */
8:              case 1: append (01)_16 to D_cmp
9:              case 2: append (02)_16 to D_cmp
10:             case 3: append (03)_16 to D_cmp
11:         append i to D_index
12:     else
13:         append d_i to D_uncmp /* 4 or 8 bytes */
14:     end if
15: end for
```

The character symbol can be coded in a single byte, e.g., ASCII code, for the linear prediction of an input original floating-point value (4 bytes for single precision and 8 bytes for double precision). We can assign different character symbols to express $2^8 = 256$ predictions at maximum in Line 6. However, since the hit ratios for $n = 0, 1, 2, 3$ are relatively high, we simply attempt the first four predictions described in Equations 1 - 4 (the detail is evaluated in Section 4.1.2.).

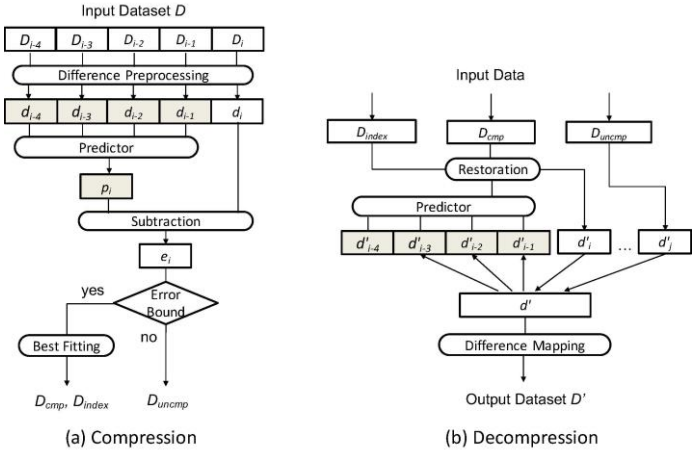


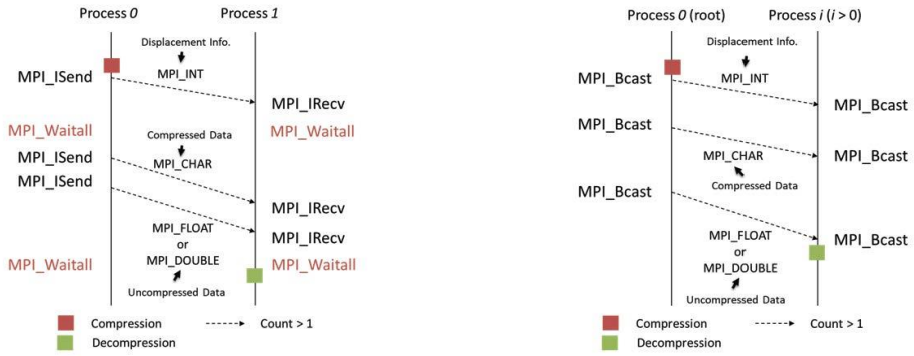
Figure 1. Block diagram of the byte-wise (de)compression for 1-D floating-point array.

Notice that D_{index} represents the displacement information of the predicted data. To identify d_i is stored in D_{cmp} or D_{uncmp}, the displacement information is needed for its smooth decompression at the receiver side. The block diagram of the byte-wise (de)compression is illustrated in Fig. 1. Since the decompression operation is performed as opposed to the compression operation, we omit to describe the decompression algorithm like Algorithm 1.

3.1.2. Implementation

We describe the implementation of the byte-wise compression for two basic MPI communication mechanisms. The first kind of communication is the point-to-point communication. The second kind of communication is the collective communication established amongst a group of processes.

MPI is a standard upon which many implementations exist. The byte-wise compression uses basic MPI functions such as MPI_Isend, MPI_Irecv and MPI_Waitall, and basic MPI data types such as MPI_INT, MPI_FLOAT/DOUBLE and MPI_UNSIGNED_CHAR, thus providing high portability to various MPI implementations.



(a) Point-to-point (b) Collective (broadcast)

Figure 2. The byte-wise compression to MPI messages.

Fig. 2(a) shows the diagram of adding the byte-wise compression to MPI point-to-point messages. This example assumes that process 0 sends messages to process 1. Because the compressed data and the uncompressed data are transferred separately, the receiver process should be aware of their respective counts to provide the corresponding receive buffers in advance. For this reason, after the sender completes data compression, it first sends the displacement information (constructed with MPI_INT) of the compressed data, which is essential for data decompression at the receiver side. The sender then transfers the compressed data (constructed with MPI_UNSIGNED_CHAR) and the uncompressed data (constructed with MPI_FLOAT or MPI_DOUBLE). After receiving the data, the receiver performs the decompression by using the displacement information. Fig. 2(b) shows the example of the implementation of adding the byte-wise compression to MPI collective (broadcast) messages. This example assumes that process 0 is the root process and it simultaneously sends messages to other processes. The flow is similar to that in the point-to-point communication.

3.2. Linear-predictive Bit-wise Compression

In this subsection, we improve the compression ratio by a bit-wise compression technique at the cost of a moderate increase of compression complexity.

3.2.1. Approximation of IEEE 754 Floating-point Values

We describe the background information on rounding IEEE 754 floating-point values. IEEE 754 standard specifies a single-precision floating-point format for a 32-bit value, which consists of 1 sign bit, 8 exponent bits and 23 mantissa bits, and a double-precision floating-point format for a 64-bit value, which consists of 1 sign bit, 11 exponent bits and 52 mantissa bits [25]. A floating-point value is computed by $sign \times mantissa \times 2^{exponent}$, where the sign is 1 or -1 if the leading bit is 0 or 1, respectively. The mantissa expresses a real value between 1.0 and 2.0, with a fractional part represented in binary format. For the single-precision format, the exponent equals the 8 bits in the middle minus 127; for the double-precision format, the exponent equals the 11 bits in the middle minus 1,023. For example, the single-precision value (0, 10000000, $10010010000111111010000)_2$ expresses $(-1)^0 \times 1.570795\ldots \times 2^{128-127} \approx (3.14159)_{10}$.

The least significant bits (LSBs) in the mantissa have little impact on the value of a floating-point number. In the bit-wise compression, we only retain the necessary b bits in the mantissa while maintaining the user-defined error bound. Thus, we aggressively discard the last $23 - b$ bits for the single-precision format, $52 - b$ bits for the double-precision format, as neglectable LSBs in the mantissa. In the above example, discarding the last 10 bits, e.g., setting the last 10 bits to 0s, will make the value 3.1413574 with 0.0074% error. If the error bound is set to 10^{-3}, then this inexact value is acceptable in terms of data precision.

It is obvious that the number of necessary bits in the mantissa, i.e., b, depends on the error bound. We figure out the value of b of the floating-point number d_i according to the predefined error bound E. First, we determine the integer value of n where $2^{-n} \leq E < 2^{-n+1}$ ($n > 0$). Then, we calculate $b = m + n$ where $2^m \leq d_i < 2^{m+1}$. Here, if $b < 0$, we set $b = 0$. The least necessary number of bits for the error-bounded compression is $1 + 8 + b = 9 + b$ for a single-precision floating-point value, and $1 + 11 + b = 12 + b$ for a double-precision floating-point value. Therefore, we can get the compression ratio, $32/(9 + b)$ for a single-precision floating-point value, and $64/(12 + b)$ for a double-precision floating-point value. Obviously, the double precision floating-point data benefits more from the lossy bit-wise compression due to a larger compression ratio.

3.2.2. Design

Algorithm 2 The bit-wise compression.

Input:
1-D floating-point array D, and user-defined error bound E
Output:
bitstream D_{bit} (encapsulated in a byte array)

```
1:  for i = 1 → M do
2:      convert D_i into non-negative floating-point d_i
3:      bestfit = arg min (|p_i^0 − d_i|, |p_i^1 − d_i|, |p_i^2 − d_i|, |p_i^3 − d_i|)
4:      if p_i^bestfit ≤ E then
5:          switch bestfit do
6:              case 0: append (100)_2 to D_bit
7:              case 1: append (101)_2 to D_bit
8:              case 2: append (110)_2 to D_bit
9:              case 3: append (111)_2 to D_bit
10:     else
11:         cut the LSBs, and append the remaining bits to D_bit
12:     end if
13: end for
```

We describe the bit-wise compression in Algorithm 2. The concept of using the linear prediction is the same as the byte-wise compression. For the smooth decompression, we convert the input floating-point values to non-negative values in Line 2 to ensure that the sign bits of negative

floating-point values are flipped to zero like in the byte-wise compression. In Line 11, we discard the maximum length of LSBs under the condition that the user-defined error bound is satisfied.

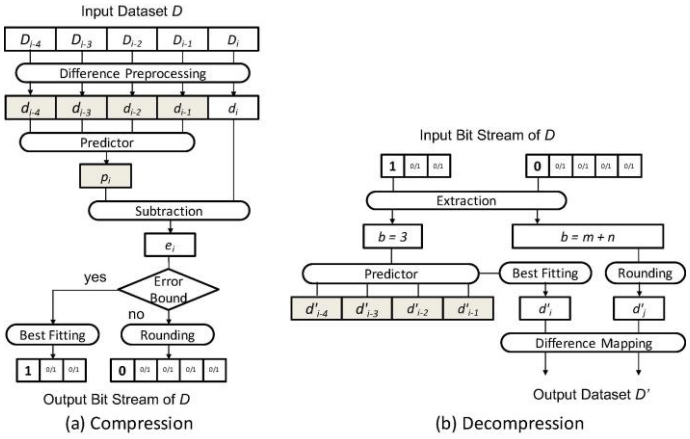


Figure 3. Block diagram of the bit-wise (de)compression for 1-D floating-point array.

Figure 3 illustrates the block diagram of the bit-wise data compression and decompression, respectively. This procedure of the linear prediction is similar to that in the byte-wise compression described in Section III-A. If the prediction fails ($p_i^{bestfit} > E$), the original value is compressed to a variable bit length, i.e., $9 + b$ bits for a single-precision floating-point value and $12 + b$ bits for a double-precision floating-point value.

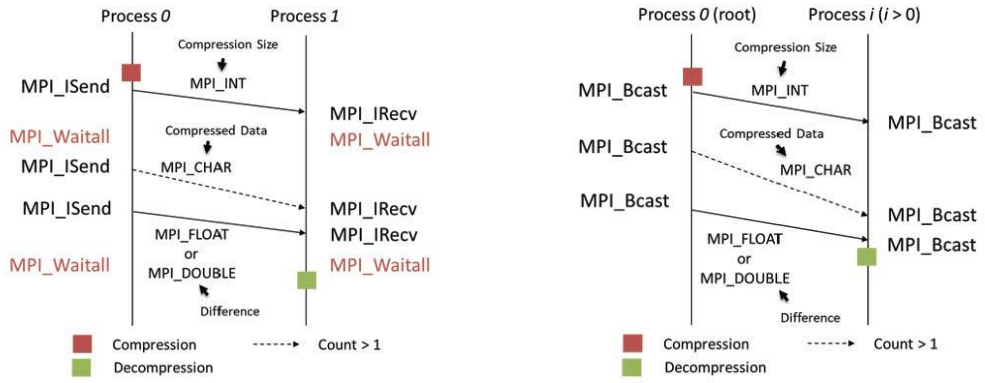
All these bits are concatenated to a continuous output bitstream. Note that the encoded data bits can be organized in the output bitstream in accordance with their original order in the input dataset. In other words, the bit-wise compression technique does not require any displacement information like in the byte-wise compression, and thus reduces the communication overhead.

For the bit-wise decompression, we can easily reconstruct the linearly predicted data and the bit-wise compressed data extracted from the received bitstream. First, we recognize the leading bit of each data piece, 1 or 0. To decode the data piece, it is also indispensable to know the bit length of the data piece, i.e., how many bits follow the leading bit. If the leading bit is 1, the data piece belongs to the linearly predicted data, thus the bit length is always 3. Then we decode the linearly predicted data by applying the calculation similar to Equations 1 - 4. Otherwise, if the leading bit is 0, the data piece belongs to the bit-wise compressed data, thus the bit length is $9 + b$ for single-precision floating-point data and $12 + b$ for double-precision floating-point data. In this case, the value of b is the key to calculate the total bit length of the data piece for its decoding. Remember that the value of b depends on the value of the exponent, thus we can get $b = m + n$, where $2^{-n} \leq E < 2^{-n+1}$ ($n > 0$) and the value of m can be obtained from the exponent bits. The lost LSBs in the data piece are padded with $(1000\ldots)_2$. For single-precision floating-point data, the number of supplemental bits is $23 - b$; for double-precision floating-point data, the number of supplemental bits is $52 - b$.

After the decoding phase, we perform a simple difference mapping procedure to convert the decoded dataset to the final decompressed dataset, which is the inverse operation of the difference preprocessing in the compression phase.

3.2.3. Implementation

The bit-wise compression uses basic MPI functions such as MPI_Isend, MPI_Irecv, and MPI_Waitall, and basic MPI data types such as MPI_INT, MPI_FLOAT/DOUBLE and MPI_UNSIGNED_CHAR for high portability to various MPI implementations.



(a) Point-to-point (b) Collective (broadcast)

Figure 4. The bit-wise compression to MPI messages.

Figure 4(a) shows the implementation of adding the bit-wise compression to MPI point-to-point messages. For the bit-wise compression, the sender only needs to send the size of the compressed data (constructed with a single MPI_INT) before transferring the compressed data (constructed with MPI_UNSIGNED_CHAR). Afterwards, the sender transfers the difference information (constructed with a single MPI_FLOAT or MPI_DOUBLE) generated at the difference preprocessing phase for decompression at the receiver side. Obviously, the bit-wise compression exchanges smaller amount of MPI communication messages when compared to the byte-wise compression. Figure 4(b) shows the implementation of adding the bit-wise compression to MPI collective (broadcast) messages.

4. EVALUATION

We firstly perform the byte- and bit-wise compression techniques on a real machine consisting of two compute nodes. We secondly perform them on an event-driven system simulator assuming a modern 64-node system. We finally investigate the impact of the compression ratio on the network throughput and communication latency using a cycle-accurate network simulator.

4.1. Parallel Application Performance on A Real Machine

4.1.1. Condition

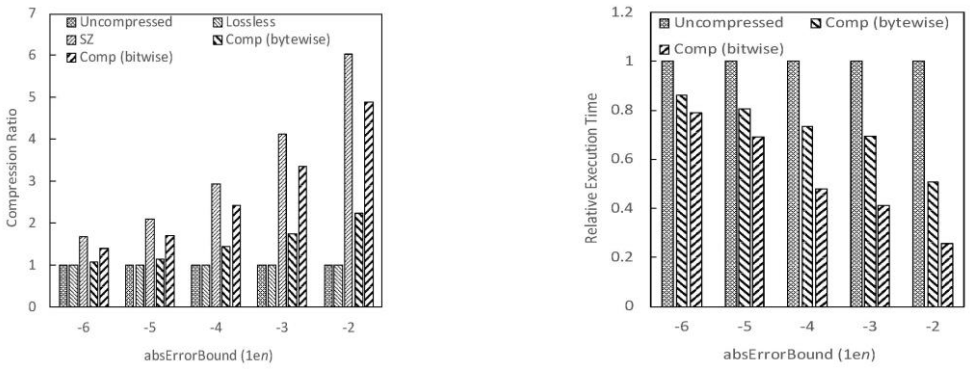
We perform the evaluation using the MPI applications running on two compute nodes in which Intel Xeon Processor X5690 is equipped with a 3.47 GHz 12-core processor. The two compute nodes have a GbE network interface, Broadcom NetXtreme II BCM5709 1000Base-T. We use OpenMPI v3.1.3 for inter-process communication on Linux Kernel 4.9.0-8-amd64. Table 1 describes the communication data types generated in the evaluated MPI applications.

Table 1. MPI parallel applications used in the evaluation.

Application	Data Type	Message Count
Ping Pong	MPI_FLOAT	8,192
Himeno	MPI_FLOAT	16,384
K-means Clustering	MPI_DOUBLE	2,366,316 (obs_info) 4,386,200 (num_plasma)
FFT	MPI_DOUBLE	2,101,248

4.1.2. Ping Pong

We introduce our floating-point data compression to a ping-pong MPI program. Two processes use MPI_Send and MPI_Recv to continually bounce messages off of each other until they decide to stop. A ping_pong_count is initiated to zero, and it is incremented at each Ping Pong step by the sending process. As the ping_pong_count is incremented, the processes take turns being the sender and receiver. Finally, after the limit is reached (10,000 in this study), the processes stop sending and receiving. We use 8,192 samplings (at time step 100) single-precision (32 bits) floating-point data from Blast2 [26] as the input dataset.



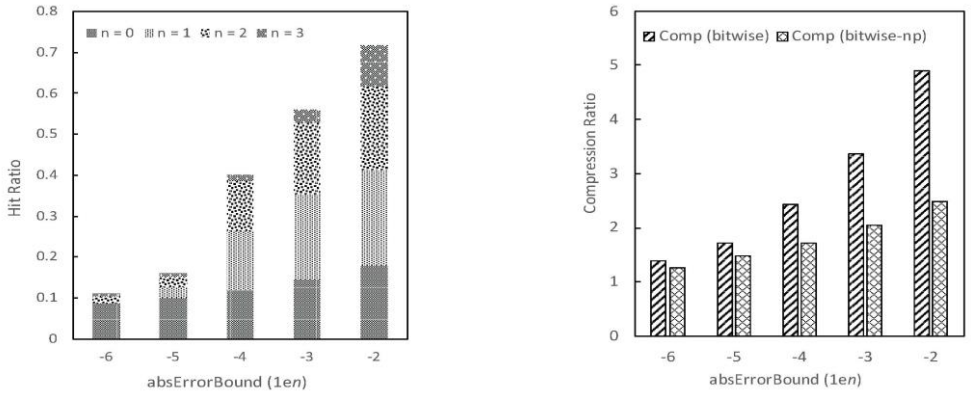
(a) Compression ratio (b) Application execution time

Figure 5. Evaluation of Ping Pong on two nodes.

Figure 5(a) shows the compression ratio with different given error bounds. Since the target data are single-precision floating-point numbers, we set the error bound E from 10^{-6} to 10^{-2}. We select Lossless [1] and SZ (v1.0) [3] as our competitors. Lossless is one of state-of-the-art lossless compression techniques. It uses a single one-dimensional polynomial predictor, subtracts from the original value, and compresses the successive zeros in their difference. SZ is one of the state-of-the-art lossy compression algorithms, as described in Section 2.1.. Unsurprisingly, the lossy compression algorithms obtain higher compression ratios than the lossless ones while maintaining within the defined error bounds. As the error bound relaxes, the compression ratio becomes larger for SZ and our compression techniques. Comparatively, the bitwise compression technique maintains a higher compression ratio than the byte-wise compression technique, and it gets comparable performance to SZ. For instance, when the error bound is 10^{-4}, the bit-wise compression algorithm gets 4.9x compression ratio, which reaches 82.6% compression ratio compared to SZ.

Figure 5(b) depicts the execution time of the Ping Pong MPI application by compressing floating-point communication data using the byte-wise and bit-wise compression techniques. We measure the execution time using the MPI_Wtime function that is inserted just before and after the target MPI functions, thus the execution time does not include the MPI initialization and finalization. Notice that SZ is not directly applicable to MPI inter-process communication on interconnection networks. We thus omit it in Fig. 5(b). The lossless compression is not included in the evaluation because its compression ratio is almost 1.0, as illustrated in Fig. 5(a).

According to the comparison with the original version of the MPI application, we found that both the byte-wise and bit-wise compression techniques obtain better performance in terms of execution time. Due to a higher compression ratio of MPI messages transferred between processes, the bit-wise compression technique performs better than the byte-wise one. For instance, when the error bound is 10^{-4}, compared to the uncompressed version, the byte-wise and bit-wise compression techniques reduce the execution time by 40.1% and 74.4%, respectively.



(a) Hit ratio of different linear predictions (b) Compression ratio w/ and w/o linear predictions

Figure 6. Effect of linear predictions on Ping Pong.

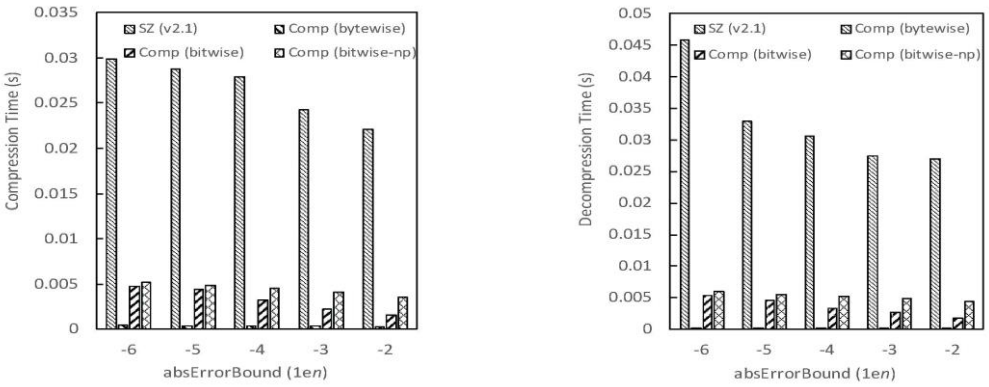
To understand the behavior of the compression techniques, Figure 6(a) illustrates the breakdown of hit ratios of the first four linear predictions (n = 0, 1, 2, 3) of the proposed compression techniques in the Ping Pong application. Notice that both the byte-wise and bit-wise compression techniques have the same hit ratio. The total hit ratio increases from 11.1% to 71.9% as the error bound relaxes from 10^{-6} to 10^{-2}. The first three linear predictions contribute much to an increase of the compression ratio. Figure 6(b) shows the comparison of compression ratios of the bit-wise compression techniques including and excluding linear predictions in the Ping Pong application. The latter (bitwise-np) applies the bit rounding to both predictable data and unpredictable data. In this case, its compression ratio is lower than the bit-wise compression technique which uses the linear predictions to compress each predictable data to only three bits. This proves that the linear prediction plays an important role in the bitwise compression technique due to its high compression ratio.

Table 2. Compound error based on error bound (bit-wise).

absErrorBound (1en)	-6	-5	-4	-3	-2
Compound Error	0	0.000001	0.000085	0.000697	0.006649

We maintain the error bound for each individual data prediction, but error can compound via compression of consecutive data pieces. Table 2 provides the results for using the bit-wise

compression technique in the Ping Pong application, which give the final compound error after the processes stop sending and receiving. It can be found that the final compound error can be still maintained within the different predefined error bounds.



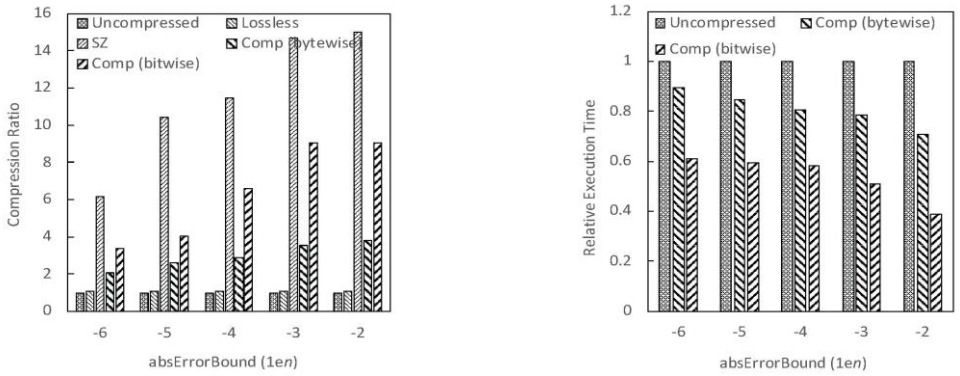
(a) Compression time (b) Decompression time

Figure 7. Compression/decompression time on Ping Pong.

We shift to compare the time cost of compression and decompression with SZ (v2.1), as shown in Fig. 7. Since SZ is not directly applicable to inter-process communication on interconnection networks, here, SZ is evaluated as a reference. Simply relying on the linear prediction, the byte-wise compression technique significantly outperforms SZ as well as the bit-wise compression technique in terms of both the compression time and decompression time. The bit-wise compression technique maintains high superiority to SZ, e.g., 8.5x speedup for compression and 9.1x speedup for decompression when the error bound is 10^{-4}, although it is inferior to the byte-wise compression technique. The bit-wise compression technique without linear predictions (bitwise-np) takes larger compression time and decompression time when compared to both the byte-wise and bit-wise compression techniques. Considering its worse compression ratio than the bitwise compression technique, the bitwise-np compression is omitted in the following evaluation.

4.1.3. Himeno

The Himeno benchmark program is developed to take measurements to proceed with major loops in solving the Poisson's equation solution using the Jacobi iteration method [27]. We modified the Himeno benchmark program for the purpose of the compression evaluation. In the Himeno benchmark program, the most used MPI functions are MPI_Isend and MPI_Irecv. The calculation size we use in the evaluation is *m* (256 × 128 × 128).



(a) Compression ratio (b) Application execution time

Figure 8. Evaluation of Himeno benchmark on two nodes.

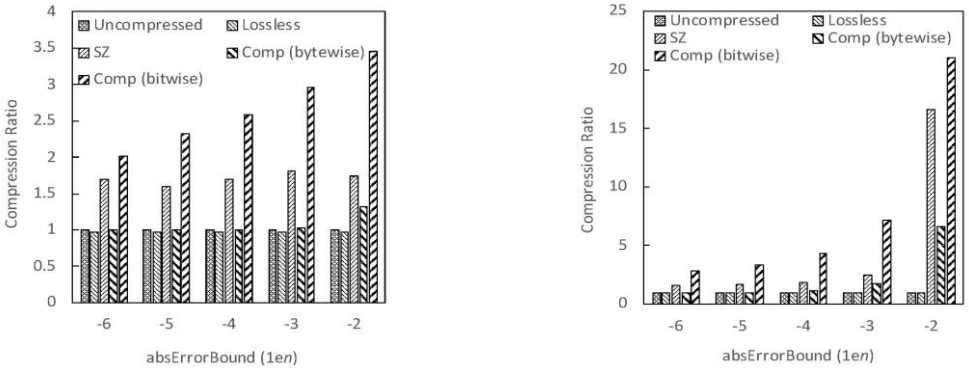
Figures 8(a) and 8(b) show the compression ratio and the application execution time, respectively. They both show a similar tendency to those in the Ping Pong application. In the case of the error bound 10^{-4}, the bit-wise compression upgrades the compression ratio by x6.6 and improves the execution time by x1.7 when compared to the uncompressed version.

4.1.4. K-means Clustering

The program of K-means clustering [28] partitions input dataset into subsets called clusters. The similar elements are placed in the same cluster. The similarity is calculated based on distance metrics, such as euclidean distance or hamming distance. In the evaluation, we assume 100 clusters and set the maximum calculation iteration to 1,000.

We employ the following two input datasets in our evaluation.

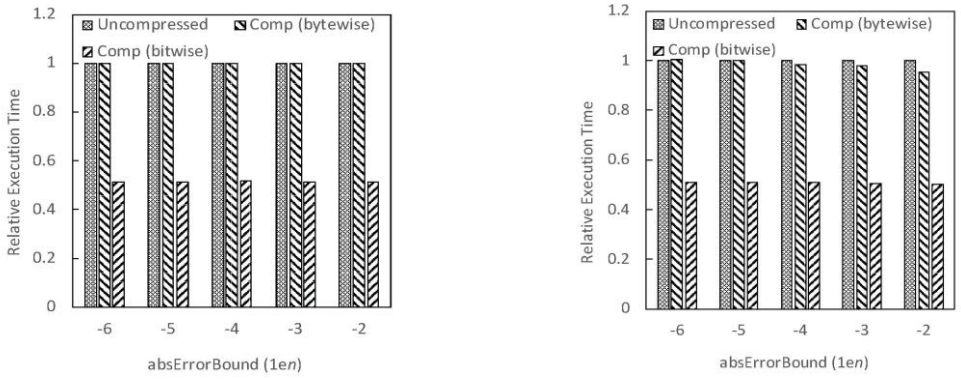
- obs_info: measurement from scientific instruments which comprises latitude and longitude information of the observation points of a weather satellite (0.3% are unique values)
- num_plasma: the result of numeric simulations which simulate plasma temperature evolution of a wire array z-pinch experiment (23.9% are unique values)



(a) obs_info (b) num_plasma

Figure 9. Compression ratio of K-means clustering on two nodes.

The evaluation results of the compression ratio are depicted in Fig. 9. A finding is that the bit-wise compression technique outperforms SZ for any error bound between 10^{-6} and 10^{-2}, because the difference preprocessing converts the original floating-point data to small values that increase the compression ratio for the bit-wise compression technique. The use of a more linear prediction (n = 3) than SZ also helps to further improve the compression ratio for the bit-wise compression technique. Besides, we found that the bit-wise compression achieves a high compression ratio as the error bound relaxes to 10^{-2}, which is very close to the theoretically maximum value for the compression of double-precision floating-point data. This implies that the linear prediction almost contributes the entire data compression, which compresses a 64-bit value to 3 bits. On the other hand, the clustering result would become more flexible when the error bound relaxes to 10^{-2}.



(a) obs_info (b) num_plasma

Figure 10. Application execution time of K-means clustering on two nodes.

As shown in Fig. 10, the bit-wise compression technique reduces the execution time by half for either obs_info or num_plasma. However, the byte-wise compression technique presents a tiny advantage when compared to the uncompressed communication data in terms of application execution time. This is because the root node repeatedly broadcasts the k-means arrays, which aggravate the frequent operations of the compression and decompression. The compression benefit for the data communication is thus sacrificed by the increased (de)compression time overhead.

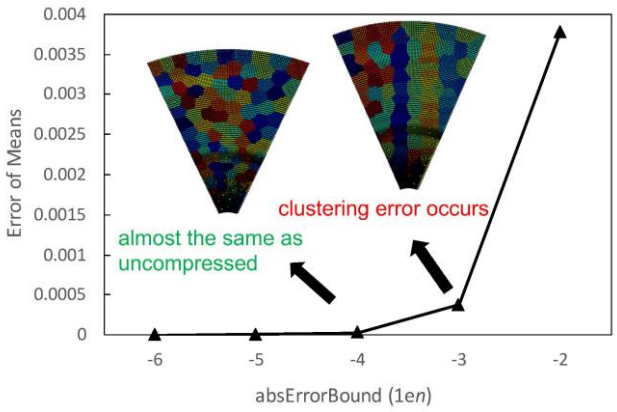
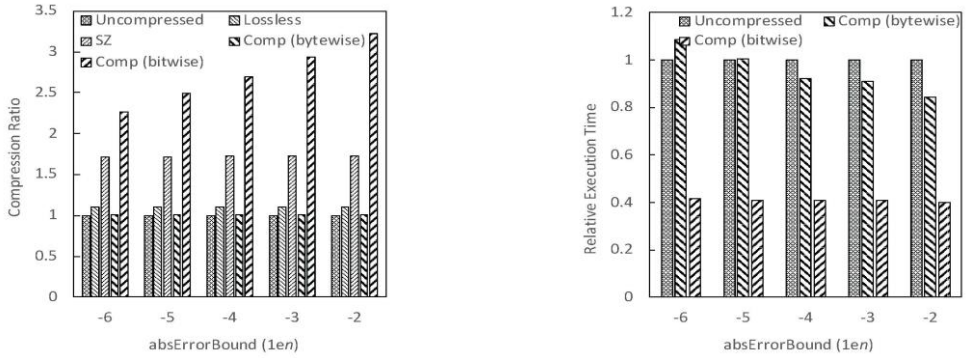


Figure 11. Error of k-means clustering (num plasma).

Figure 11 shows the average error of means on the clusters for the dataset num_plasma when using the bit-wise compression technique, together with its impact on the final clustering result. The same color indicates the same cluster in the clustering output. When the error bound relaxes to 10^{-4}, the bit-wise compression still generates almost the same clustering result as the uncompressed version. The clustering error occurs when the error bound jumps over 10^{-3}.

4.1.5. FFT

The program of FFTSS [29] is an open source library for computing the Fast Fourier Transform (FFT). The FFTSS library includes various FFT kernel routines. We modify FFTSS v3.0 to implement the floating-point data compression techniques. In the evaluation, we create and execute a plan for computing the double-precision data with two-dimensional transforms for processors with MPI library.



(a) Compression ratio (b) Application execution time

Figure 12. Evaluation of FFT on two nodes.

The evaluation results of compression ratio and execution time are depicted in Fig. 12(a) and 12(b), respectively. The byte-wise compression seems not beneficial for both the measurements because it simply relies on the linear prediction, which seems not to work well in this case. The bit-wise compression significantly outperforms the byte-wise compression as the error bound relaxes. It keeps a higher compression ratio than SZ within the error bounds from 10^{-6} to 10^{-2} and saves the execution time by around 60% compared to the uncompressed version.

4.2. Parallel Application Performance on 64 Compute Nodes

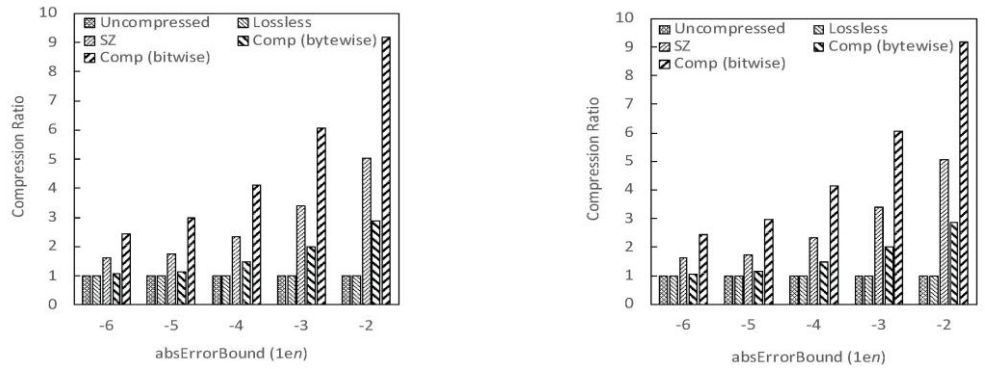
4.2.1. Condition

Since we do not have a large-scale real machine, instead, we use a discrete-event simulation to evaluate the performance of parallel application benchmarks. To this end, we use the SimGrid (v3.21) simulation framework [30]. It simulates the executions of the unmodified MPI parallel applications [31] and the modified versions using the byte- and bit-wise compression techniques.

In the simulation, we assume the minimal routing using the Dijkstra algorithm on a 4×4×4 3-D torus interconnection network. We configure SimGrid so that each switch has a 100ns delay, switches and compute nodes are interconnected via the links with 200Gbps bandwidth each, and each compute node has a 5TFlops computation power. The parameters of each application are the same as those in the previous subsection.

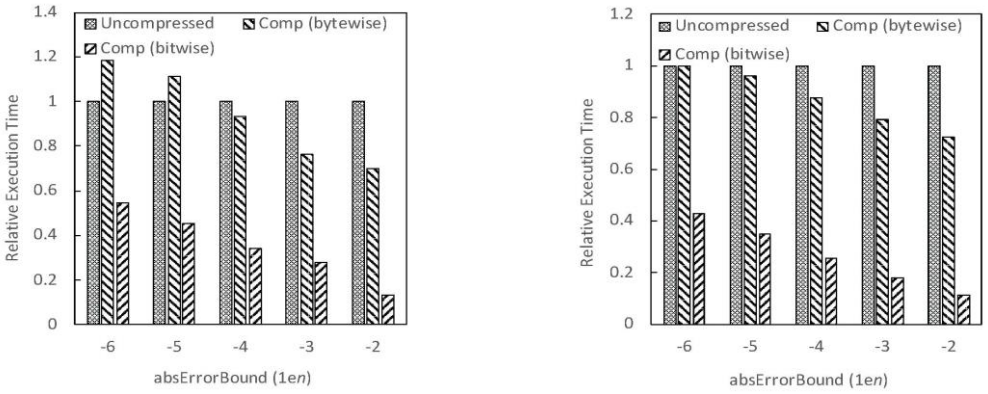
4.2.2. Ping Pong

We simulate the synthetic traffic patterns that determine each source-and-destination communication node pair: random uniform and matrix-transpose. Each process exchanges the same dataset as that used in the previous subsection, according to the synthetic access patterns. These traffic patterns are commonly used for measuring the performance of interconnection networks, as described in [32]. A node injects data packets into the interconnection networks independently of each other.



(a) Random uniform (b) Matrix-transpose

Figure 13. Compression ratio of Ping Pong on 64 nodes.

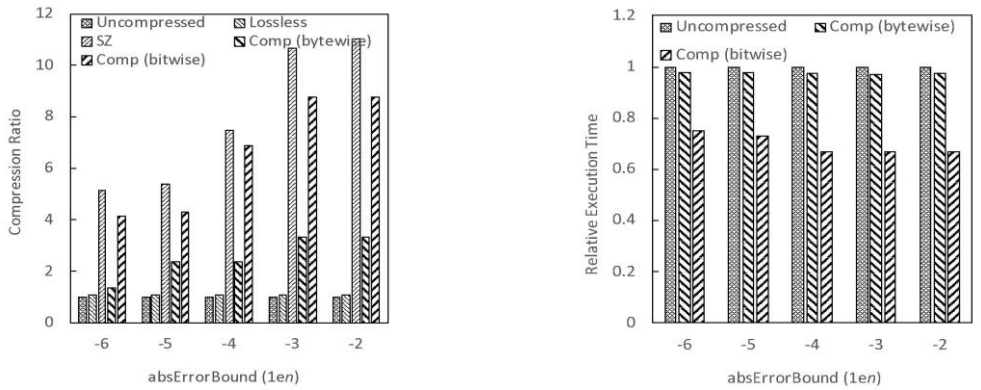


(a) Random uniform (b) Matrix-transpose

Figure 14. Relative execution time of Ping Pong on 64 nodes.

Figure 13 illustrates the comparison of the compression ratio, and Figure 14 is the execution time relative to the non-compression communication on the interconnection network. These results are consistent with the results in Fig. 5(a) and 5(b). We observe that the improvement ratios over the original uncompressed interconnection network become high as the acceptable quality turns low.

4.2.3. Himeno

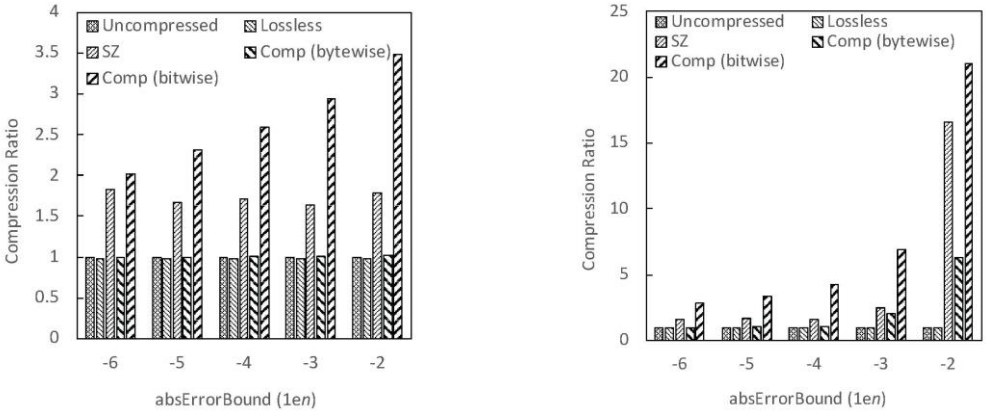


(a) Compression ratio (b) Application execution time

Figure 15. Evaluation of Himeno benchmark on 64 nodes.

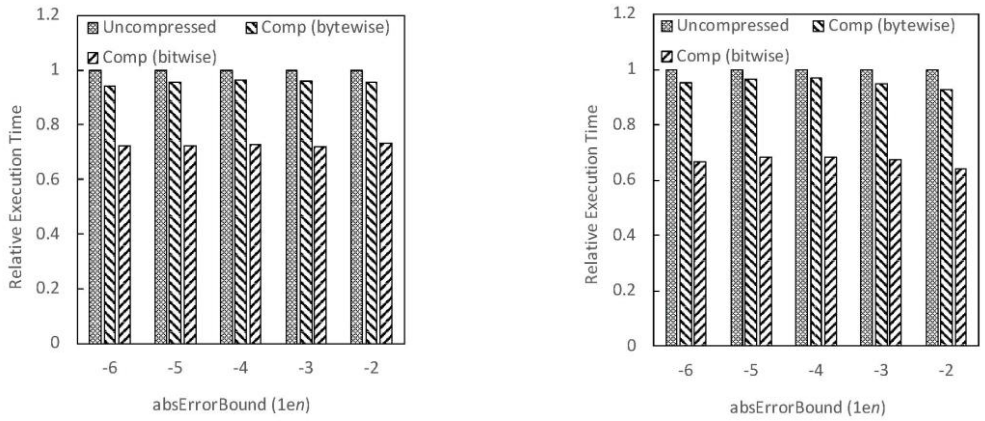
We show the simulation results for the Himeno benchmark in Fig. 15(a) and 15(b). We set the calculation size as *s* (128 × 64 × 64) and tuned the iteration times according to the simulated CPU speed, which do not impact the comparison of our data compression algorithms. In this case, the byte-wise compression technique seems to bring a limited improvement in terms of either compression ratio or execution time. This indicates that the linear prediction plays an insignificant role in the dataset for the byte-wise compression technique. By contrast, the bit-wise compression technique obtains a comparable compression ratio to SZ, and reduces the execution time by up to 33.2% compared to the original uncompressed version.

4.2.4. K-means Clustering



(a) obs_info (b) num_plasma

Figure 16. Compression ratio of K-means clustering on 64 nodes.



(a) obs_info (b) num_plasma

Figure 17. Application execution time of K-means clustering on 64 nodes.

As depicted in Fig. 16 and 17, we perform the evaluation of k-means clustering using the datasets obs_info and num_plasma. The settings of other parameters are the same as those in the real-machine evaluation. Similarly, the trends indicate the superiority of the bit-wise compression technique using both the datasets. More specifically, due to the benefit of small values obtained by the difference preprocessing phase, the bit-wise compression technique presents a higher compression ratio than SZ especially as the error bound relaxes, and it reduces the execution time by around 30%. Similarly to the evaluation on a real machine, the bit-wise compression technique achieves a high compression ratio close to the theoretically maximum value when the error bound relaxes to 10^{-2}.

4.3. Throughput and Latency

We illustrated the performance of MPI applications in the previous subsection. In this subsection, we generalize and investigate the effective network performance obtained by different compression ratios using a cycle-accurate network simulation.

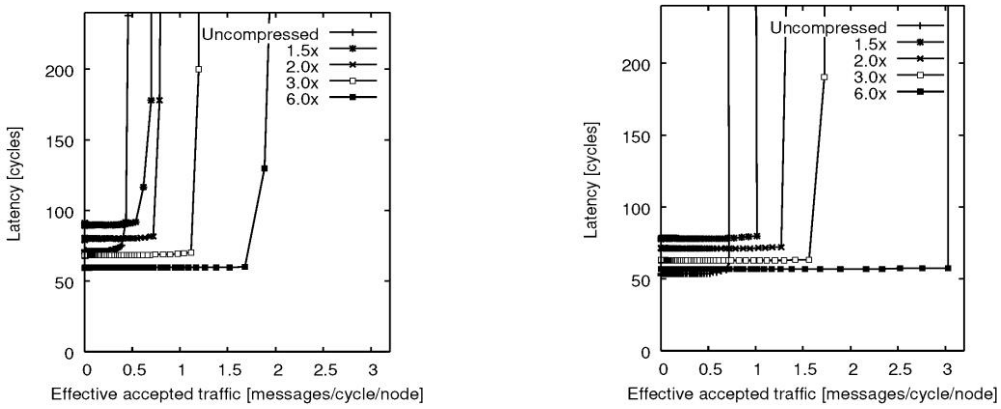
4.3.1. Condition

We use a cycle-accurate network simulator written in C++ [33]. A router model consists of channel buffers, a crossbar and a link controller, and the control circuits are used to simulate the switching fabric. On a conventional packet router, a header flit transfer requires four cycles that include routing, virtual-channel allocation, switch allocation and flit transfer from an input channel to an output channel through a crossbar. We use the minimal adaptive routing via two virtual channels on a 4×4×4 3-D torus interconnection network. Each node includes a router with a local processor. The packet length is set to 32 flits when no data compression is applied. We simulate the same synthetic traffic patterns as those in the previous subsection: random uniform and matrix-transpose.

We investigate the impact of the compression ratio of data packets on the communication latency and the effective network performance. Although the compression ratio depends on the data type and compression algorithm, we parameterize the compression ratio for illustrating the network performance in this simulation. For the consistency with the previous subsection, the compression ratio is set to 1.5, 2.0, 3.0 and 6.0. We set 40 cycles as the (de)compression overhead at each source and destination pair.

Our results show two important metrics: communication latency and effective throughput. The communication latency is the elapsed time between the generation of a packet at a source host and its delivery at a destination processing element. We measure the communication latency in simulation cycles. The effective throughput is defined as the maximum accepted traffic represented by the flit delivery rate. The flit delivery rate is computed as $C \times R$, where C and R are the compression rate and the average number of received flits at a processing element within a cycle.

4.3.2. Results



(a) Random uniform (b) Matrix-transpose

Figure 18. Communication latency vs. accepted traffic for (non-)compression communications.

Figure 18 illustrates the relationship between the communication latency and the accepted traffic rate for non-compression and compression communication datasets. The plot of "2.0x" represents the case where all packets are compressed with the compression ratio of 2.0.

As the compression ratio increases, the packet length becomes shorter, thus efficiently reducing the network injection latency. Since the communication latency includes the network injection latency, we observe that a higher compression ratio leads to a better communication latency at the low traffic load. For example, the 1.5 and 2.0 compression ratios bring the higher communication latency than no compression on the interconnection network at the low traffic load. By contrast, the 3.0 and 6.0 compression ratios improve the communication latency even at the low traffic load with the random uniform traffic.

A higher compression ratio also leads to a better effective network throughput, since shortening the packet length by the data compression decreases the network load. For example, when the compression ratio is 3.0, it improves the effective network throughput by 176% and 140% with the traffic patterns of random uniform and matrix transpose, respectively. Similarly, the 1.5 and 6.0 compression ratios improve the effective network throughput by up to 133% and 260%, respectively.

Through the simulation results, we observe that a high compression ratio improves both the communication latency at a high traffic load and the effective network throughput with both the evaluated traffic patterns.

5. CONCLUSION

Data compression increases the effective network bandwidth on an interconnection network of parallel computers. Generally, a lossy compression achieves a higher compression ratio than that by a counterpart lossless compression. In this study, we introduce a lossy compression algorithm to floating-point communication data on interconnection networks.

Since recent interconnection networks are latency-sensitive, a simple lossy compression technique that has small compression overhead is preferred. In this context, we apply a linear predictor with the user-defined error bound to floating-point communication data compression. We provide byte- and bit-wise compression techniques. In the byte-wise compression, if the value prediction succeeds, a floating-point value is converted to a single byte expression, corresponding to an MPI char type. Otherwise, the original value is not compressed. Although the byte-wise compression has low compression-operation latency, its upper bound of the compression ratio is not high, i.e., obviously four and eight for single-precision and double-precision floating-point values, respectively. By contrast, the bit-wise compression generates a bitstream encapsulated in a byte array, corresponding to an MPI char type. If the value prediction succeeds at a source node, the floating-point value is converted to three bits. Even if the value prediction fails, the least significant bits (LSBs) are discarded from the IEEE 754 floating-point expression of the value in order to obtain a relatively high compression ratio, while maintaining a given error bound.

We implemented and evaluated the byte- and bit-wise compression techniques for floating-point communication data generated in the MPI parallel programs of Ping Pong, Himeno, K-means Clustering and FFT. The bit-wise compression technique achieves 2.4x, 6.6x, 4.3x and 2.7x compression ratio for Ping Pong, Himeno, K-means and FFT at the cost of a moderate decrease of quality of results (error bound is 10^{-4}), thus achieving 2.1x, 1.7x, 2.0x and 2.4x speedup of the execution time, respectively. More generally, from the network point of view, the cycle-accurate network simulation shows that, when the compression ratio becomes 1.5, 3.0 and 6.0, the interconnection network improves the effective network throughput by up to 133%, 176% and 260%, respectively. Through this observation, we highly recommend using the lossy application-level compression, i.e., the bit-wise compression, on interconnection networks.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 19H01106 and JST PRESTO JPMJPR19M1.

REFERENCES

[1] T. Ueno, K. Sano, and S. Yamamoto, “Bandwidth compression of floating-point numerical data streams for fpga-based high-performance computing,” ACM Transactions on Reconfigurable Technology and Systems, vol. 10, pp. 1–22, 05 2017.
[2] J. Tomkins, “Interconnects: A Buyers Point of View,” ACS Workshop, 2007.
[3] S. Di and F. Cappello, “Fast error-bounded lossy hpc data compression with sz,” in IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2016, pp. 730–739.
[4] M. Narasimha and A. Peterson, “On the computation of the discrete cosine transform,” IEEE Transactions on Communications, vol. 26, no. 6, pp. 934–936, June 1978.
[5] D. H. J. Michael T. Heideman and C. S. Burrus, “Gauss and the history of the fast fourier transform,” IEEE ASSP Magazine, vol. 1, no. 4, pp. 14–21, 1984.
[6] C. C. Cutler, “Differential quantization of communication signals, u.s. patent 2605361,” July 1952.
[7] L. Deng and D. O’Shaughnessy, “Speech processing: a dynamic and optimization-oriented approach,” in Marcel Dekker, 2003, pp. 41–48.

[8] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," IEEE transactions on visualization and computer graphics, vol. 12, pp. 1245–50, 09 2006.
[9] P. Lindstrom, "Fixed-rate compressed floating-point arrays," IEEE Transactions on Visualization and Computer Graphics, vol. 20, 08 2014.
[10] S. Lakshminarasimhan, N. Shah, S. Ethier, S.-H. Ku, C. Chang, S. Klasky, R. Latham, R. Ross, and N. Samatova, "Isabela for effective in situ compression of scientific data," Concurrency and Computation: Practice and Experience, vol. 25, 02 2013.
[11] M. Isenburg, P. Lindstrom, and J. Snoeyink, "Lossless compression of floating-point geometry," in Computer-Aided Design and Applications, vol. 1, 04 2004.
[12] M. Isenburg, P. Lindstrom, and J. Snoeyink, "Lossless compression of predicted floating-point geometry," Computer-Aided Design, pp. 869–877, 07 2005.
[13] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pp. 1245–1250, 2006.
[14] P. Ratanaworabhan, J. Ke, and M. Burtscher, "Fast lossless compression of scientific floating-point data," in Data Compression Conference Proceedings, 04 2006, pp. 133– 142.
[15] M. Burtscher and P. Ratanaworabhan, "High throughput compression of double-precision floating-point data," in Data Compression Conference (DCC), 2007, pp. 293–302.
[16] M. Burtscher and P. Ratanaworabhan, "Fpc: A high-speed compressor for double-precision floating-point data," Computers, IEEE Transactions on, vol. 58, pp. 18 – 31, 02 2009.
[17] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization," in IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2017, pp. 1129–1139.
[18] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo, Z. Chen, and F. Cappello, "Error-controlled lossy compression optimized for high compression ratios of scientific datasets," in IEEE International Conference on Big Data (Big Data), 2018, pp. 438–447.
[19] X. Liang, S. Di, S. Li, D. Tao, B. Nicolae, Z. Chen, and F. Cappello, "Significantly improving lossy compression quality based on an optimized hybrid prediction model," in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC, 2019, pp. 33:1–33:26.
[20] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka, "Exploration of lossy compression for application-level checkpoint/restart," in IEEE International Parallel and Distributed Processing Symposium, 2015, pp. 914–922.
[21] A. R. Alameldeen and D. A. Wood, "Frequent pattern compression: A significance-based compression scheme f or l2 caches," in Technical Report 1500, Computer Sciences Dept. UW-Madison, Apr. 2004.
[22] R. Das, A. K. Mishra, C. Nicopoulos, D. Park, V. Narayanan, R. R. Iyer, M. S. Yousif, and C. R. Das, "Performance and power optimization through data compression in network-on-chip architectures," in International Conference on High-Performance Computer Architecture (HPCA), 2008, pp. 215–225.
[23] B. Dickov, M. Peric`as, P. M. Carpenter, N. Navarro, and E. Ayguad´e, "Analyzing performance improvements and energy savings in infiniband architecture using network compression," in 2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing, Oct 2014, pp. 73–80.
[24] V. Engelson, D. Fritzson, and P. Fritzson, "Lossless compression of high-volume numerical data from simulations," in Proceedings of the Data Compression Conference, 02 2000, pp. 574–586.
[25] "Ieee standard for floating-point arithmetic," IEEE Std 754-2019 (Revision of IEEE 754-2008), pp. 1–84, 2019.
[26] P. Colella and P. R. Woodward, "The piecewise parabolic method (ppm) for gas-dynamical simulations," Journal of Computational Physics, vol. 54, no. 1, pp. 174 – 201, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999184901438
[27] "Himeno benchmark," http://i.riken.jp/en/supercom/documents/himenobmt/.
[28] "k-means clustering: A distributed mpi implementation," https://github.com/dzdao/k-means-clustering-mpi.
[29] A. Nukada, "Fftss: A high performance fast fourier transform library," in IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 3, 2006, pp. III–III.
[30] "Simgrid: Versatile simulation of distributed systems," http://simgrid.gforge.inria.fr/.

[31] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, “Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms,” Journal of Parallel and Distributed Computing, vol. 74, no. 10, pp. 2899–2917, 2014.
[32] J. Duato, S. Yalamanchili, and L. Ni, Interconnection Networks: an engineering approach. Morgan Kaufmann, 2002.
[33] A. Jouraku, M. Koibuchi, and H. Amano, “An Effective Design of Deadlock-Free Routing Algorithms Based on 2-D Turn Model for Irregular Networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 3, pp. 320–333, Mar. 2007.
[34] Q. Fan, D. J. Lilja and S. S. Sapatnekar, "Using DCT-based Approximate Communication to Improve MPI Performance in Parallel Clusters," 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), London, United Kingdom, 2019, pp. 1-10.

AUTHORS

Yao Hu received the M.S. degree from Beijing University of Posts and Telecommunications, China, in 2009, and received the PhD degree from the Department of Computer Science and Engineering, Waseda University, Tokyo, Japan, in 2015. He is currently working as a project researcher in the National Institute of Informatics, Tokyo, Japan. His main research interests include the area of high-performance computing.

Michihiro Koibuchi received the BE, ME and PhD degrees from Keio University, Yokohama, Japan, in 2000, 2002 and 2003, respectively. Currently, he is an associate professor in the Information Systems Architecture Research Division, National Institute of Informatics and the Graduate University of Advanced Studies, Tokyo, Japan. His research interests include the areas of high-performance computing and interconnection networks. He is a senior member of the IEICE, IEEE and IPSJ.

A STUDY OF IDENTIFYING ATTACKS ON INDUSTRY INTERNET OF THINGS USING MACHINE LEARNING

Chia-Mei Chen[1], Zheng-Xun Cai[1], Gu-Hsin Lai[2]

[1]Department of Information Management,
National Sun Yat-sen University, Taiwan
[2]Department of Technology Crime Investigation,
Taiwan Police College, Taiwan

ABSTRACT

The "Industry 4.0" revolution and Industry Internet of Things (IIoT) has dramatically transformed how manufacturing and industrial companies operate. Industrial control systems (ICS) process critical function, and the past ICS attacks have caused major damage and disasters in the communities. IIoT devices in an ICS environment communicate in heterogeneous protocols and the attack vectors might exhibit different misbehavior patterns. This study proposes a classification model to detect anomalies in ICS environments. The evaluation has been conducted by using ICS datasets from multiple sources and the results show that the proposed LSTM detection model performs effectively.

KEYWORDS

Industry Internet of Things, Machine Learning, Anomaly Detection.

1. INTRODUCTION

The emerging technologies of Industry Internet of Things (IIoTs) and 5G have started a new chapter for industries and businesses. The "Industry 4.0" revolution has dramatically transformed how manufacturing and other industrial companies operate. Industry 4.0 converges operations technology (OT) and information technology (IT) networks. While this union of these formerly disparate networks certainly facilitates data exchange and enables organizations to improve business efficiency, it also comes with a host of new security concerns [1].

Industrial Control System (ICS) is a major core system for manufacturing processes and critical infrastructure operations, which is a collection of all control systems, such as Supervisory Control and Data Acquisition (SCADA), Industrial Automation and Control System (IACS), Distributed Control System (DCS), Process Control System (PCS) [2]. It plays a significant role in industrial physical infrastructure to ensure each fundamental element to be under surveillance and co-work with others.

Industry control systems (ICS) and other OT devices used to be deployed in an internal network without security protection but are connected to IT networks nowadays. ICS design had not considered the security functionality, but now ICS, sensors, and other controllers become IP-enabled and IIoT endpoints on the converged OT/IT network, which expand the attack surface and increase security risk for enterprises.

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 77-82, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110707

IIoT devices and ICS systems have become a primary attack target based on their importance to business operation and national security. ICS attacks became prevailing in this decade, ranging from power plants, gas pipelines, water cleaning, energy and petrochemical companies, financial sectors. In 2010, Stuxnet was the first known threat targeting specific ICS systems, followed by Dugu, Flame, and Gauss targeting specific ICS manufacturers in 2011 [3]. The attacks are expected to increase in number and sophistication. Therefore, critical infrastructure owners and operators must develop the ability to detect and recover from cyberattacks [4]. Therefore, protecting ICS and IIoT networks is critical for enterprises and industries.

Traditional IDS mostly work on a signature basis, and there are not many known signatures to detect attacks on ICS networks [5]. IDS for IT networks might not work sufficiently in such an emerging OT/IT environment, because ICS components and IIoT devices behave differently from IT devices.

This study proposes a classification method to detect anomalies in heterogeneous network environments. In the first phase, it identifies the network protocol of the traffic and detects anomalies in the second phase by applying the LSTM model, where LSTM is an artificial neural network architecture with feedback connections that can process single data points as well as sequences of data, such as time series traffic flows.

The remainder of this paper is constructed as follows. Section 2 reviews the previous related research. Section 3 presents the proposed detection method, followed by the performance evaluation in Section 4. The last section draws the conclusion remark and the future directions of this study.

2. LITERATURE REVIEW

A study [6] highlighted common characteristics of IoT networks, in which a multitude of different devices with different capabilities and different communication protocols communicate with each other. It categorizes IoT attacks by different network layers: physical, link, network, transport, and application layers. Another study [7] defined a taxonomy model for attacks on most used industrial communication protocols: Modbus and DNP3.

Some research identified anomalies based on the recurring network patterns of IIoT networks. A study [5] analyzed the network traffic from a water distribution system in order to understand how ICSs work and evaluated the off-the-self IDS solutions. It concludes that network traffic from an ICS is static and the off-the-self IDS solutions produced high false positives. A study [8] developed an open-source PLC for validating PLC logic execution and comparing PLC behaviors when under an injection attack with crafted packets. Another study [9] designed an approach that exploits traffic periodicity to detect traffic anomalies by sliding windows but also pointed out that changes in the traffic periodicity are not necessarily malicious.

A study [10] developed an anomaly event detection model for a water system controlled by ICS and evaluated the following six ML algorithms: logistic regression, Gaussian naïve Bayes, k-nearest neighbors, support vector machine, decision tree, and random forest. The experimental results discovered that the more recorded attack scenarios used for training, the more robust the detection model.

Neural Network (NN) is one of the most popular ML algorithms, which have been employed on anomaly detection. Radford et al. [11] showed that RNN can represent sequences of communications on a network and discover anomalous network traffic. Prasse et al. [12] analyzed HTTPS network flows, employed a natural language model to extract features from domain

names, and proposed an LSTM-based detection method. Their experimental results show that the LSTM classification model outperforms a random forest model. Kim and Ho [13] employed CNN to extract spatial features and LSTM temporal characteristics and proposed a neural network for detecting anomalies on web traffic.

3. PROPOSED METHODOLOGY

According to the literature review on the major ICS attack events [4], most of the attacks have involved abnormal network behaviors. On the other hand, the literature review also has demonstrated MLtechniques yield effective classification and LSTM is suitable for time-series data. Therefore, this study analyzes the ICS traffic and employs an LSTM detection model to identify anomalous traffic.

The IIoT devices on the converged OT/IT network communicate in multiple communication protocols; therefore, attack scenarios might be different based on the applied protocol. Our preliminary study discovers that the machine-learning model performs better on identifying anomalies of a single protocol than on identifying anomalies of different protocols.

To improve detection effectiveness, this study proposes a detection method that applies the LSTM classification model to identify malicious traffic. The raw network traffic in packets is captured from the network, and the module Preprocess merges packets into flows, where a flow is a sequence of packets from a source to a destination. The detail of the proposed method is explained below.

3.1. Protocol Categorization

The past research applied ML models [14-16] to identify network protocols, which requires intensive computation and training time. Although IIoT devices may communicate in heterogeneous network protocols, it is reasonable to assume that the protocols applied within an IIoT network environment are known so that the security control of network monitoring can be operated. Given the knowledge of the protocol formats and the valid range of each field in the headers, the module Protocol Categorization identifies the protocol by examining the header.

3.2. Feature Encoding

Feature encoding is a process of transforming a categorical variable, such as protocol type, into a continuous variable. The payload (Data segment) is represented in hex, and its entropy is included in the feature set.

The selected feature set consists of relevant features from the applied communication protocol. For the protocols over TCP/IP, the following TCP/IP features are included: source IP, destination IP, port, packet size. For a specific ICS communication protocol, all its header fields, payload, and payload entropy are included in the feature set. Transaction ID, Protocol ID, the size of the payload, Unit ID, Function Code, and the payload and its entropy are extracted.

3.3. Detection Model

ICS processes perform periodic tasks; therefore, the communication contents among the IIoT devices are stable and periodical. The LSTM is suitable for learning time series and periodic patterns according to the literature review. The loss function adopted is binary cross-entropy as it is a binary classifier, where cross-entropy loss increases as the predicted probability diverges

from the actual label and penalizes misclassification greatly to build a good detection model. Most model learning optimizers are based on the stochastic gradient descent technique. RMSprop [17] uses an adaptive learning rate, instead of treating the learning rate as a hyper-parameter, and is suitable for learning a model from a big or redundant dataset. The training data of this study contains repeated traffic patterns that fit the above description. Therefore, this study adopts RMSprop for model optimization.

4. EVALUATION AND DISCUSSION

This research adopts accuracy, precision, recall, and F1 score as performance measurements. Accuracy is the proportion of correct predictions among the total number of the examined cases; precision as known as positive predictive value is defined as the fraction of the true instances among the predicted instances; recall as known as sensitivity is the fraction of the total number of the correctly identified instances divided by the total number that were predicted to be positive instances; F1 score that combines precision and recall is the harmonic mean of precision to compare the two models in this study.

This study evaluates the proposed solution with the datasets from multiple sources [18-20] and divides them into training and testing with the ratio of 8:2. The dataset source [18] contains multiple datasets collected from power systems, a gas pipeline and water storage tank, and gas pipelines. The dataset CSET [21] was collected from an electrical network monitor system that consists of two MTUs and three SCADA controllers. The 4SICS Geek Lounge [20] contains an ICS lab with PLCs, RTUs, servers, and industrial network equipment (routers, switches, firewalls, network cameras).

4.1. Experiment : Effectiveness of the Classification Method

This experiment evaluates the performance of the proposed detection method that categorizes the network protocol of the traffic data and then identifies anomalies. Table 1 presents the results of the LSTM detection model with and without protocol categorization. The classification model achieves better performance, as attack patterns might be different on different protocols.

Table 1. The detection results with and without protocol categorization.

	LSTM
Accuracy	86.58%
Precision	95.81%
Recall	64.84%
F1	0.773371722

5. CONCLUSION

This study proposes a ML-based classification model to identify anomalies on heterogeneous IIoT network environments and demonstrates that the impact of imbalanced training data and the resampling on the detection performance. The proposed a detection method first categorizes the protocol types of traffic flows and then identifies the anomalies of each protocol type. Such an approach reduces the training time as well as improves the detection efficiency on the heterogeneous network environments.

Most past research has focused on improving performance by evaluating different ML models. Besides that, this study demonstrates that upsampling improves the efficiency of model learning and the basic unit of data resampling may affect detection performance.

The multi-layered IT/OT converged network environments expand the attack surface; hence, defense-in-depth [22] is recommended by implementing layers of defense mechanisms. This study identifies the attacks of the two low levels on the ICS environments. While targeted attacks become very stealthy and customized, future work can extend the detection by cross-correlating security alerts from different layers of ICS environments.

REFERENCES

[1] R. Best. "Converged OT/IT Networks Introduce New Security Risks." https://www.infosecurity-magazine.com/opinions/ot-it-networks-risks/ (accessed: 20 Dec., 2020).

[2] O. KOUCHAM, "Détection d'intrusions pour les systèmes de contrôle industriels." [Online]. Available: https://tel.archives-ouvertes.fr/tel-02108208/document

[3] K. E. Hemsley and E. Fisher, "History of industrial control system cyber incidents," Idaho National Lab.(INL), Idaho Falls, ID (United States), 2018.

[4] K. Hemsley and R. Fisher, "A history of cyber incidents and threats involving industrial control systems," in International Conference on Critical Infrastructure Protection, 2018: Springer, pp. 215-242.

[5] J. Angséus and R. Ekbom, "Network-based intrusion detection systems for industrial control systems," Master Thesis, Department of Computer Science and Engineering, Chalmers University of Technology, 2017.

[6] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: current solutions and future challenges," IEEE Communications Surveys & Tutorials, 2020.

[7] Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of attacks on industrial control protocols," in 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), 2015: IEEE, pp. 1-6.

[8] T. Alves and T. Morris, "OpenPLC: An IEC 61,131–3 compliant open source industrial controller for cyber security research," Computers & Security, vol. 78, pp. 364-379, 2018.

[9] R. R. R. Barbosa, R. Sadre, and A. Pras, "Towards periodicity based anomaly detection in SCADA networks," in Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012: IEEE, pp. 1-4.

[10] H. Hindy, D. Brosset, E. Bayne, A. Seeam, and X. Bellekens, "Improving SIEM for critical SCADA water infrastructures using machine learning," in Computer Security: Springer, 2018, pp. 3-19.

[11] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," arXiv preprint arXiv:1803.10769, 2018.

[12] P. Prasse, L. Machlica, T. Pevný, J. Havelka, and T. Scheffer, "Malware detection by analysing network traffic with neural networks," in 2017 IEEE Security and Privacy Workshops (SPW), 2017: IEEE, pp. 205-210.

[13] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," Expert Systems with Applications, pp. 66-76, 2018.

[14] C. Jeong, M. Ahn, H. Lee, and Y. Jung, "Automatic Classification of Transformed Protocols Using Deep Learning," in International Conference on Parallel and Distributed Computing: Applications and Technologies, 2018: Springer, pp. 153-158.

[15] J. Xue, Y. Chen, O. Li, and F. Li, "Classification and identification of unknown network protocols based on CNN and T-SNE," in Journal of Physics: Conference Series, 2020, vol. 1617, no. 1: IOP Publishing, p. 012071.

[16] R. Lin, O. Li, Q. Li, and Y. Liu, "Unknown network protocol classification method based on semi-supervised learning," in 2015 IEEE International Conference on Computer and Communications (ICCC), 2015: IEEE, pp. 300-308.

[17] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," vol. 14, no. 8.

[18] U. Adhikari, S. Pan, and T. Morris. "Industrial Control System (ICS) Cyber Attack Datasets." https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (accessed: 18 Dec., 2020).

[19] A. Lemay. "A SCADA Dataset." https://github.com/antoine-lemay/Modbus_dataset (accessed: 18 Dec., 2020).
[20] 4SICS Geek Lounge. "Capture files from 4SICS Geek Lounge." https://www.netresec.com/?page=PCAP4SICS (accessed: 18 Dec., 2020).
[21] A. Lemay and J. M. Fernandez, "Providing SCADA network data sets for intrusion detection research," in 9th Workshop on Cyber Security Experimentation and Test, 2016.
[22] A. Fielder, T. Li, and C. Hankin, "Defense-in-depth vs. critical component defense for industrial control systems," in 4th International Symposium for ICS & SCADA Cyber Security Research 2016 4, 2016, pp. 1-10.

A NEW HASHING BASED NEAREST NEIGHBORS SELECTION TECHNIQUE FOR BIG DATASETS

Jude Tchaye-Kondi, Yanlong Zhai and Liehuang Zhu

School of Computer Science, Beijing Institute of Technology, Beijing, China

ABSTRACT

KNN has the reputation of being a simple and powerful supervised learning algorithm used for either classification or regression. Although KNN prediction performance highly depends on the size of the training dataset, when this one is large, KNN suffers from slow decision making. This is because each decision-making process requires the KNN algorithm to look for nearest neighbors within the entire dataset. To overcome this slowness problem, we propose a new technique that enables the selection of nearest neighbors directly in the neighborhood of a given data point. The proposed approach consists of dividing the data space into sub-cells of a virtual grid built on top of the dataset. The mapping between data points and sub-cells is achieved using hashing. When it comes to selecting the nearest neighbors of a new observation, we first identify the central cell where the observation is contained. Once that central cell is known, we then start looking for the nearest neighbors from it and the cells around. From our experimental performance analysis of publicly available datasets, our algorithm outperforms the original KNN with a predictive quality as good and offers competitive performance with solutions such as KDtree.

KEYWORDS

Machine learning, Nearest neighbors, Hashing, Big data.

1. INTRODUCTION AND MOTIVATIONS

The K nearest neighbors or simply KNN is an algorithm that relies on a very simple principle: tell me who your neighbors are and I'll tell you who you are(Figure 1). Therefore, to make a prediction, KNN does not rely on a statistical model, it learns nothing from the training data and has to carry the full dataset during its decision making. For this reason, KNN is categorized as a Lazy Learning algorithm.

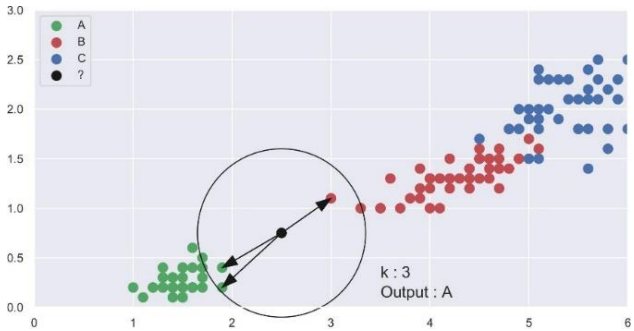


Figure 1: KNN example

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 83-95, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110708

The K of KNN is not a parameter but a hyperparameter because, unlike conventional parameters, it will not be learned automatically by the algorithm from the training data. It is up to us to optimize it, using the test dataset. To generate a prediction for a new observation x, the algorithm searches for the K closest instances to x from the dataset. For these instances, the algorithm will be based on their target values to generate the output of the observation we want to predict. Accordingly:

- If we are in a regression task, the prediction is the average or median of the K nearest instances' labels.
- If we are in a classification task, the prediction is commonly majority class among the K nearest instances' labels.

Figure 1 illustrates simple classification example with k = 3. We can clearly observe that among the 3 nearest data points, 2 belongs to class A. KNN will then output class A for the black data point. KNN is a supervised learning method with very good prediction accuracy, hence its wide use in several domains. In medicine, researchers proposed a KNN based drug classification approach used to categorize the different types of drug[1], it is also used for diagnosing Heart disease patients[2], cancer prediction and detection[3][4][5]. In computer vision, KNN work efficiently in image classification [6], face recognition [7]. The KNN algorithm is also present in cybersecurity where it is effectively used for credit card fraud detection[8], detection of intrusive attacks in a network system[9]. The strength of KNN is its simplicity and efficiency. However, behind this efficiency, it hides two big weaknesses that are:

- The large size of the final model: since KNN is not a statistical model, all training datasets must be carried out during inferences.
- Slow inferences: during inference, KNN must iterate through the dataset for distance calculations before selecting the nearest neighbors. This process has a time complexity of $(fn)log(fn)$ with n the training dataset size and f the number of features.

In addition, KNN is also very sensitive to noise (outliers), highly dependent on the choice of$K$ and the distance metric. The slowness and model size weaknesses restrict the use of KNN with large training data. Since KNN runs in memory and as the RAM is limited, it will be difficult to keep a large dataset in memory. Due to the slowness that results, this algorithm is not suitable for real-time applications or applications having strict time limits requirements. In this paper, we focused on the slowness problem of KNN during predictions with big datasets. To improve the prediction time efficiency, we are proposing a new hashing-based algorithm called GHN: Grid Hashing Neighborhood. GHN approach consists of splitting the data space into sub-cells by building a virtual grid on top of it. The virtual grid is constructed in such a way that each data point in the dataset can be mapped to a specific grid cell with a hash function. Both the virtual grid and the mapping hash function are constructed at the learning phase and then used during inferences to speed up the nearest neighbors selection. During prediction, the nearest neighbors of a new observation are selected in two steps. First, the central cell to which the new observation belongs is identified using the mapping hash function. Second, we search for nearest neighbors from this central cell and cells around it layer by layer. Therefore, unlike the native KNN, which have to go through the entire training dataset, GHN is able to select the nearest neighbors directly in the neighborhood of a new observation. Our performance analysis shows that our approach is faster in making predictions than the native KNN.

The rest of the paper is structured as follows. Section 2 reviews related work. Section 3 the methodology of GHN. Section 4 evaluates the performance of our implementation of GHN, the

original KNN, and KDtree on some publicly available datasets. Finally, the paper is concluded in Section 5.

2. RELATED WORK

The slowness of KNN predictions is not a new problem in machine learning, unlike humans who, just by looking at the data representation in 2D or 3D vector space, can intuitively guess the nearest neighbors of a data point, a computer requires more calculations for the same task. Three main groups emerge among the different techniques used to compute the nearest neighbors that are: data reduction approaches, hashing based approaches, and tree-based approaches. Most of the literature's suggested solutions consist of data reduction strategies. These strategies try to reduce the size of the training data, therefore, reducing the number of distance calculations and the amount of memory needed by the model during prediction. Reducing the size of the training data can be effective for certain types of datasets that may still work accurately by only considering some special data points. Although they are quicker in prediction and enhance memory usage, these approaches are less adopted. It may be because data reduction usually does not lead to the same prediction accuracy as KNN.

In [10],[11],[12],[13] concave and convex hulls-based techniques are proposed and used to reduce each class samples to their edge data points. In these techniques, only the edge points are used in the training datasets for classification. Hart et Al. proposed the condensed nearest neighbor(CNN)[14] which reduces its data by selecting prototypes U from the training data in a way that 1NN with U can classify the samples almost as precisely as 1NN does with the dataset. CNN works in 3 steps [15]: 1) Scans all the elements of the training data X, looking for an element x whose nearest prototype from U has a label different from x. 2) Remove x from X and add it to U. 3) Repeat the operation until no other prototype is added to U. In the end use U instead of X to train the model. In the same perspective, Salvador et Al. introduce compressed kNN[16] which is a binary level data compression technique. The method proposes to compress observations into packets of a certain number of bits. In each packet, attributes are stored through binary level operations. This technique reduces the amount of RAM needed to maintain the training data in memory. An interesting feature of the compressed kNN approach is that the information can be decompressed, observation by observation on the fly and in real-time, without the need to decompress the entire dataset in memory. Unfortunately, compressed kNN also suffers from slowness and only works with categorical features.

During our research, we noticed that there were only a few researches attempts to use hashing techniques to estimate the nearest neighbors. Hashing can be used to group similar data points in buckets. The most popular hashing based solution is the LSH(Locality Sensitive Hashing) family [17] [18] [19] [20]. LSH based solutions use random plane projections in the data space to divide that space into sub-regions. These sub-regions are then used as a bucket to build a hash function. Even if LSH based solutions improve prediction time, the strategy behind them is ineffective and does not guarantee to get the real nearest neighbors hence its low adoption. Gao et Al. try to face this problem by suggesting another family of hashing technique, DHT[21], which, unlike the LSH family, can maintain relationships between the nearest neighbors. Tree-based solutions are the most adopted in real-world problems when it comes to approximating the nearest neighbors. The most famous are KD tree[22][23] and Ball Tree[24][25]. They are data structures that organize training data like a tree. When searching for the nearest neighbors, we navigate the tree from top to bottom, hoping that the region we led in will contain the nearest neighbors. Just like with LSH, these tree-based solutions can easily miss the real nearest neighbors and they are mostly recommended for low dimensional space since they don't perform well with multi-dimensions. In conclusion, there is still no efficient solution to accurately estimate the KNN that

provides low computation and memory cost. The existing one suffers from drawbacks like accuracy degradation or the risk of having fake nearest neighbors. Our solution adopts a unique hashing-based approach that allows us to directly select our neighbors around the observation during prediction with relatively good performances.

3. PROPOSED SOLUTION

Compared to the other machine learning algorithms KNN does not have a learning phase, the dataset does not undergo any transformation and is entirely maintained in memory. It is during predictions that KNN does all of its computations (distances, nearest neighbors selection). Unlike KNN, GHN has a learning phase before the prediction one. Figure 2 illustrates the GHN algorithm. We work with the simple case of a two-dimensional space, i.e. when the training data only have two features since it is much easier to visualize and understand. However, it can be generalized to multiple dimensions. Figure 2.a contains the observations of two classes, the class A observations are the blue squares, that of class B are the green circles, and the new observation whose class is to be predicted is represented by a red cross. For this example, $k = 3$, so our goal is to find the 3 nearest data points to our new observation. GHN algorithm consists mainly of two steps:

1) **Cells sampling:** this phase is accomplished during training. We subdivide the data space into identical sub-cells by building on top of it a virtual grid as illustrated in Figure 2.b. A Hash function is used to map training data points to their corresponding sub-cells.
2) **Exploration:** this phase consists of selecting the nearest neighbors of a given input. As shown in Figure 2.c, GHN firstly determines the sub-cell to which the new observation with unknown output belongs by using the mapping hash function. Secondly, it searchesnearest neighbors from data points in this central cell and cells in its neighborhood layers by layer.

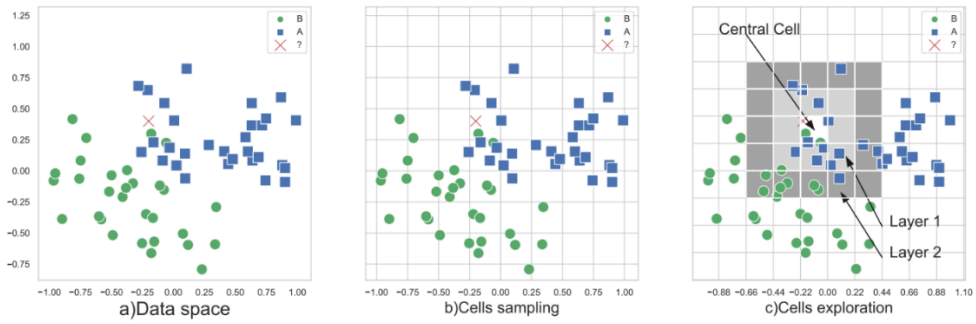


Figure 2: Different steps off GHN algorithm

3.1. Cell sampling

It is done during the model's training phase. As said above, during this phase, we build a virtual grid on top of the data space to split it into sub-cells. Each sub-cell will contain the data point located in the area it covers. These cells represent the buckets of our hash table, they are identical but not necessarily equilateral. Before building the virtual grid, we start by deciding the cell measurements on each dimension. To do this, we divide the values range covered by the training data points on each dimension in the largest possible number of splits. The division has to be done in a way that each of the splits we obtain ends up with at least one data point. The cell measurements on the corresponding dimension will then be the split width. The cell sampling process is illustrated with Figure 3, in which the first dimension can have a maximum of 7 splits

and a maximum of 8 splits for the second dimension. The cell measurements will be respectively $(range1/7, range2/8)$ on the first and second dimensions. This way of determining the virtual grid's cell measurements ensures an optimal distribution of the data points in cells, it also facilitates the lookup of nearest neighbors during exploration. Once the measurements for each dimension are determined, Equation 1 defines the hash function that maps a data point to its corresponding cell. The result of Equation 1 uniquely identifies each cell of the virtual grid. Data points that belong to the same cell have the same cell id. GHN hash table does not keep any information about empty cells for memory efficiency. The entire cell sampling process is simplified by Algorithm 1.

Algorithm 1: Cell sampling Algorithm

```
grid : A hash table where the key is cellId;
a : cell measurements;
data : training data;
/* Model Training : Cells sampling          */
for point in data do
    cellId = point//a;
    if cellId not grid then
        /* Initialize the cell              */
        grid[cellId] = [];
    end
    grid[cellId].append(point);
end
```

$$cell\_id = P//a \tag{1}$$

Where:

- P: The data point.
- $//$: Integer division.
- a: Cell Measurements

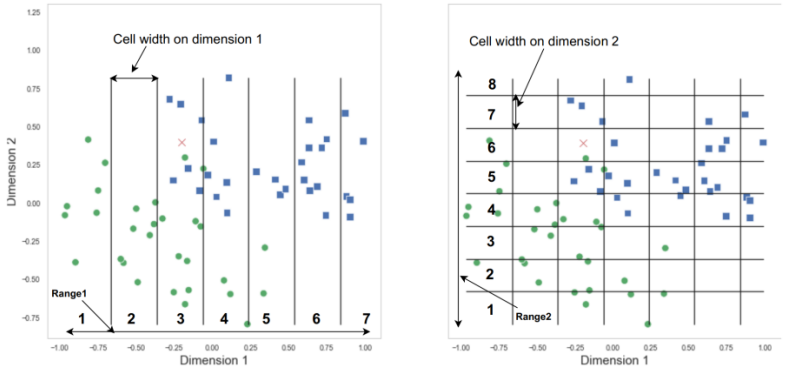


Figure 3: Cell measurements

3.2. Exploration

The actual selection of neighbors data points is made during this exploration step. The idea is to start with the cell to which our new observation belongs. Using Equation 1, we can get its id, compute the ids of cells in its neighborhood to search for the nearest data points. The two steps of the exploration are as follows:

1) Get the central cell, the one that contains the new observation using Equation 1.
2) Retrieve data points from the central cell and its neighbors layer by layer, as shown in the example of Figure 2.c.

We use the breadth-first search [26] (BFS) techniques in our implementation to compute ids and visit cells around the central one. When a cell is visited, the data points it contains are collected in a buffer. The buffer here is a heap of size k and only keeps the k potential nearest element using heap sort mechanism[27]. The exploration stops when a layer is visited and there is no update among the buffer elements. In Figure 2.c, $K = 3$, the three nearest data points are selected as follows:

- Central cell exploration: The central cell only contains 1 data point. We add this observation to our buffer, and we explore the cells on the first layer since the buffer size is less than 3.
- First layer Exploration: After visiting these 8 cells, from the data point collected from them, the buffer will only retain the 3 closest ones to our observation.
- Second layer Exploration: the second layer consists of 16 cells. Visiting these cells does not provide any update among the data points in the buffer, so the exploration stops there.

At the end of the exploration, the buffer will remain with the exact k elements that are our nearest neighbors. It is important to note that in some rare cases, the exploration may miss the real nearest neighbors due to the fact the virtual grid cells don't have the same measurements for each dimension. The entire exploration process is illustrated by Algorithm 2.

Algorithm 2: Exploration Algorithm

```
grid : The hash table ;
a : cell measurements;
data : training data;
/* Select K Nearest: Cells exploration      */
central_cellId = new_input//a;
queue = [central_cellId];
buffer = [];
while queue is not empty do
    cellId = queue.dequeue()
    if cellId not grid then
        buffer.addAll(grid[cellId])
    end
    Add neighbors' cells to the queue;
    for next_cell in neighborsCells do
        queue.enqueue(next_cell)
    end
end
return buffer;
```

3.3. Performances Comparison with KNN

The complexity of GHN mainly depends on the number of features and the exploration depth (visited layers) that is strongly influenced by the data distribution. GHN can achieve record performance with large training datasets compared to existing solutions. The more the data grows, the faster is GHN since it directly searches for neighbors in the observations' neighborhood. Its performance is almost constant $O(1)$ when the input is located in a densely populated area of the data space, otherwise, GHN will require a little more effort and more exploration. The time taken by GHN to make a prediction is the time taken by its exploration phases, added to the time needed to process the buffer that contains the k nearest elements:

$$Time\ Complexity = Exploration + Buffer\ processing$$

- Exploration time: Depends on the exploration depth and the number of data points processed from visited cells. Knowing the exploration depth, Equation 3 can help to define the total number of visited cells.
- Buffer processing time: it takes $O(k)$ time to process the buffer because only k elements remain at the end of the exploration phase.

During the exploration, the number of cells on each layer given by the following Equation 2, is proved in Appendix A.

$$n = (2l + 1)^f - (2l - 1)^f \quad (2)$$

Where:

- l: the number of visited layers.
- f: the number of dimensions or features.

Therefore, if the exploration stops after l layers, the total number of cells from the central cell to the last layer is expressed by the following Equation 3. We also demonstrate it in Appendix B.

$$S_{cells}(l) = (2l + 1)^f - 1 \quad (3)$$

The number of features f is fixed and does not change while using the model, only the exploration depth l influences $S_{cells}(l)$. It is difficult to extrapolate the different values of the exploration depth and the number of data points processed during the exploration phase since they depend on the dataset. For this reason, it is difficult to compare GHN directly with the KNN. Nevertheless, we will rather look at the best-case comparison of them. The best case that GHN allows is when the exploration is done in a very dense area and stop after exploring a single layer ($depth = l = 1$) and collecting k data points. For this best-case scenario, the number of visited cells is calculated by setting $l = 1$ in Equation 3 gives $S_1 = 3^f - 1$.Then the best-case complexity of is:

$$O(3^f - 1) + O(k) \simeq O(1) + O(1) \simeq O(1)$$

- $O(3^f - 1)$ is the cell exploration complexity.f can be neglected since it is a constant: $O(3^f - 1) \simeq O(1)$.
- $O(k)$ is the complexity for processing buffer. k also is a constant and can be neglected.

The proposed approach can achieve the best-case complexity of almost $O(1)$, which is far better than the original KNN and which has not yet been possible with all the solutions proposed so far. With original KNN, the best and worst-case time complexity is $O(nflog(fn)))$. Both GHN and KNN have a memory complexity of $O(fn)$. Although GHN uses slightly more memory than KNN to store its hash table, the number of cells will never exceed the size of the training dataset n. GHN is the only proposed solution whose prediction time performance is not negatively influenced by the growth in the training data size. With existing solutions, the more the dataset size increases the slower they are but with our approach, the more the learning data increases, the better.

3.4. Discussion

The proposed solution can be used for classifications as well as for regressions tasks as it is only intended to improve the selection of nearest neighbors. In the draft of Figure 4 we can notice that once the nearest neighbors selected, they are used as input of a classification model that is responsible for predicting the majority class by a vote or a regression model that will predict the average or the median of the k selected samples. Features in the training dataset may take their values from completely different scales of magnitude. It is recommended for the training data to be rescaled in order to set features on the same magnitude. This is carried out using data scaling techniques such as Standardization, Mean Normalization, Unit Vector, etc. GHN is not suitable for all types of datasets as it assumes that all data points converge in the same area of the data space. When the data points are far away from each other or when our observation is very far from the region where the data points are located, GHN's performance degrades. As KNN, GHN is also subject to the curse of dimensionality [28]. This occurs when the number of states exponentially increases for a tiny increase in the number of dimensions or parameters due to a combinatory explosion. The phenomena can be observed in Equation 3. Each time the number of layers increases, the number of cells to explore is an exponential function of the dimensions.

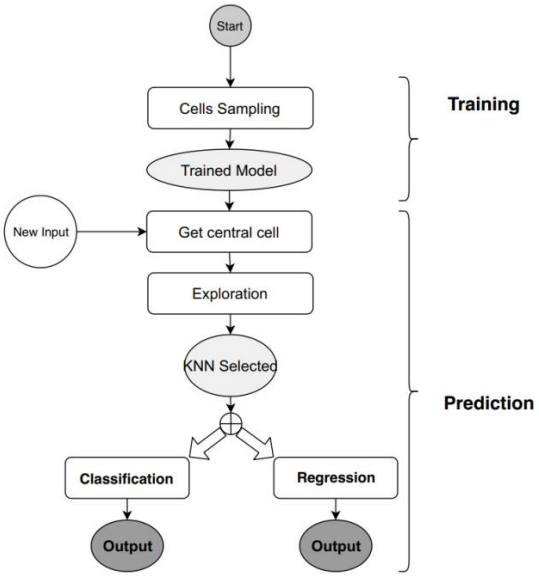


Figure 4: GHN Flow

4. EXPERIMENTS AND RESULTS

We evaluate the GHN performance against the original KNN and the popular KDTree. Our experiment target two main metrics, the prediction time which helps to evaluate the time efficiency, and the accuracy score which can tell us about how well the model is performing. Our tests are performed on 5 real dataset that we grab from different datasets repository [29], [30], [31], [32], [33], [34], these datasets are presented in Table 1. Each dataset is used for classification tasks only in order to facilitate comparisons. We scale the datasets and reduce their dimensions by using Principal component analysis(PCA). Our experiment is implemented in python 3.6 and the running environment is an Ubuntu laptop of processor Intel i7.

Table 1: Datasets

Datasets	Description
Fashion MNIST [34]	Fashion-MNIST is a dataset of Zalando's article images. Each example is a 28x28 grayscale image, associated with a label from 10 classes.
MNIST [30]	MNIST database of handwritten digits has a training set of 60,000 examples. The digits have been size-normalized and centred in a fixed-size image.
Pulsar Star [29], [33]	Describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey.
Wine Quality [31]	Data about various chemical combination of red wine.
Russian Demography [32]	Russian Demography (1990-2017) Dataset. It contains demographic features like natural population growth, birth rate, population, etc.

We have collected in Table 2 the time taken by each model to evaluate the test data for each dataset.

Table 2: Prediction Times(ms) on various datasets

Datasets	GHN	KDTree	KNN
Fashion MNIST	84.29	52.40	228.77
MNIST	10.00	5.22	111.48
Pulsar Star	9.09	15.00	60.20
Wine Quality	0.27	0.87	2.46
Russian Demography	0.19	0.90	2.92

For each dataset, 80% is used for training and the remaining 20% for testing. In the results of Figure 5, GHN and KDtree are much faster than the original KNN on the 5 datasets. We also notice that KDT is slightly faster than GHN on image data type, this is due to the effect of the curse of dimensionality faced by GHN during the exploration since a flattened image end up with a high dimensional vector. This problem is mitigated by using PCA to bring the dimensions to a good balance between speed and accuracy. On the contrary, for other types of data, GHN is faster than KDtree. By analysing Table 2 data and Figure 5 results, GHN is the best choice for real-time applications if we must choose between GHN and KNN.

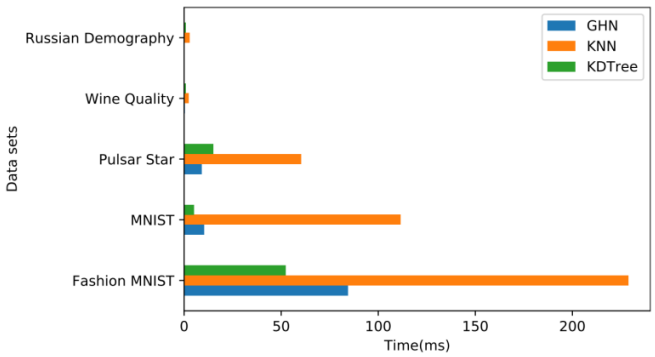


Figure 5: Comparison of prediction times on various datasets

In Table 3 and Figure 6, we compare accuracies of GHN, KDTree and KNN on all datasets.

Table 3: Accuracies on various datasets

Datasets	GHN	KDTree	KNN
Fashion MNIST	0.86	0.80	0.88
MNIST	0.74	0.74	0.74
Pulsar Star	0.99	0.99	0.99
Wine Quality	0.65	0.65	0.65
Russian Demography	0.50	0.49	0.52

With these results, we can conclude that all solutions accuracies are almost identical except for some slight variations. GHN offers better accuracy than KDtree on Russian Demography data and Fashion MNIST.

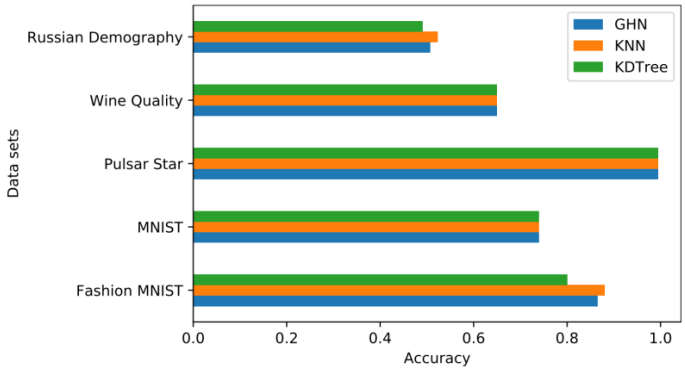


Figure 6: Comparison of accuracies on various datasets

Our experiment confirmed the fact that GHN can improve the time efficiency during predictions. It displays an accuracy as concurrent as that of KNN and KDTree. The main goal of GHN is to improve the prediction time by selecting the nearest neighbors directly in the neighborhood of our observation. Like KNN, GHN is sensitive to noise and is also subject to the curse of dimensionality. On the other hand, compared with proposed solutions till now that deal with the slowness of KNN, GHN is the only one capable of achieving almost constant time performance when the observations are in a densely populated area of the data space.

5. CONCLUSION

This paper proposes a new technique for picking the nearest neighbors that improve the prediction time for KNN, which turns out to be very slow. The algorithm works in two steps: Cell sampling and Exploration. During the first step, a hash function maps the data points with cells of a virtual grid built on top of the data space. Finally, the second step running during predictions consists of exploring and selecting the nearest data points. The experimental results validate the superior speed of GHN against KNN on all our testing datasets. GHN is compatible for both regression and classification tasks with a prediction efficiency as good as that of KNN. However, because the curse of dimensionality effect that exponentially increases the number of cells to explore from layer to layer, GHN is only recommended for low dimensional data spaces.

APPENDIX A

PROOF OF EQUATION 2

How many cells are on a given layer l? To answer this question,let's define by:

- a : cell measurements
- l : the layer
- f : the number of features

By looking at the Figure 2.c, we can deduce that the area from the central cell to a layer l is:

$$A(l) = [(2l + 1)a]^f$$

To only obtain the area covered by cells on layer l only, we must, therefore, deduct from $A(l)$ the area $A(l - 1)$:

$$A(l) - A(l-1) = [(2l + 1)a]^f - [(2(l - 1) + 1)a]^f$$

Now we can compute the number of cells on layer l. For that we just have to divide $A(l) - A(l - 1)$ by the cell volume a^f:

$$N_{cells} = \frac{A(l) - A(l - 1)}{a^f}$$

$$N_{cells} = \frac{[(2l + 1)a]^f - [(2(l - 1) + 1)a]^f}{a^f}$$

$$N_{cells} = (2l + 1)^f - (2l - 1)^f$$

Hence Equation 2.

APPENDIX B

PROOF OF EQUATION 3

In this Appendix, we want to find the number of cells from the first layer to a layer l.
We obtain the area from the first to layer l by deducing from $A(l)$ defined in Appendix A the volume of the central cell:

$$A(l) - a^f = [(2l + 1)a]^f - a^f$$

Dividing this volume with the cell area gives us the total number of cells:

$$S_{cells}(l) = \frac{A(l) - a^f}{a^f}$$

$$S_{cells}(l) = \frac{[(2l + 1)a]^f - a^f}{a^f}$$

$$S_{cells}(l) = (2l + 1)^f - 1$$

Which gives Equation 3.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their insightful suggestions. This research work is supported by the Natural Key R&D of China (No.2018YFC0830104).

REFERENCES

[1] J. Akhil, B. Deekshatulu, and P. Chandra, "Classification of heart disease using k- nearest neighbor and genetic algorithm," Procedia Technology, vol. 10, p. 85–94, 12 2013.

[2] M. Shouman, T. Turner, and R. Stocker, "Applying k-nearest neighbour in diagnosing heart disease patients," International Journal of Information and Education Technology, vol. 2, pp. 220–223, 01 2012.

[3] S. A. Medjahed, T. A. Saadi, and A. Benyettou, "Breast cancer diagnosis by using k-nearest neighbor with different distances and classification rules," International Journal of Computer Applications, vol. 62, no. 1, 2013.

[4] K. Machhale, H. B. Nandpuru, V. Kapur, and L. Kosta, "Mri brain cancer classification using hybrid classifier (svm-knn)," in 2015 International Conference on Industrial Instrumentation and Control (ICIC), May 2015, pp. 60–65.

[5] M. Jabbar, "Prediction of heart disease using k-nearest neighbor and particle swarm optimization," Biomed. Res, vol. 28, no. 9, pp. 4154–4158, 2017.

[6] G. Amato and F. Falchi, "knn based image classification relying on local feature similarity," in Proceedings of the Third International Conference on SImilarity Search and APplications. ACM, 2010, pp. 101–108.

[7] H. Ebrahimpour and A. Kouzani, "Face recognition using bagging knn," in International Conference on Signal Processing and Communication Systems (ICSPCS'2007) Australia, Gold Coast, 2007, pp. 17–19.

[8] V. R. Ganji and S. N. P. Mannem, "Credit card fraud detection using anti-k nearest neighbor algorithm," International Journal on Computer Science and Engineering, vol. 4, no. 6, pp. 1035–1039, 2012.

[9] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," Computers & security, vol. 21, no. 5, pp. 439– 448, 2002.

[10] W. M. Getz and C. C. Wilmers, "A local nearest-neighbor convex-hull construction of home ranges and utilization distributions," Ecography, vol. 27, no. 4, pp. 489–505, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10. 1111/j.0906-7590.2004.03835.x

[11] Z. Szymanski and M. Dwulit, "Improved k-nearest neighbor´ classifier for biomedical data based on convex hull of inversed set of points," in Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2010, R. S. Romaniuk, Ed., vol. 7745, International Society for Optics and Photonics. SPIE, 2010, pp. 324 – 331. [Online]. Available: https://doi.org/10.1117/12.873054

[12] A. Moreira and M. Santos, "Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points." 01 2007, pp. 61–68.

[13] J.-S. Park and S.-J. Oh, "A new concave hull algorithm and concaveness measure for n-dimensional datasets," Journal of Information Science and Engineering, vol. 29, pp. 379–392, 03 2013.

[14] P. Hart, "The condensed nearest neighbor rule (corresp.)," IEEE transactions on information theory, vol. 14, no. 3, pp. 515–516, 1968.

[15] k-nearest neighbors algorithm. https://en.wikipedia.org/wiki/Knearest neighbors algorithm. Accessed: 2020-1-10.

[16] J. Salvador, Z. Ruiz, and J. Garcia, "Compressed knn: K-nearest neighbors with data compression," 2019.

[17] J. Pan and D. Manocha, "Fast gpu-based locality sensitive hashing for k-nearest neighbor computation," in Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, 2011, pp. 211–220.

[18] Y.-M. Zhang, K. Huang, G. Geng, and C.-L. Liu, "Fast knn graph construction with locality sensitive hashing," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2013, pp. 660–674.

[19] S. Bagui, A. K. Mondal, and S. Bagui, "Improving the performance of knn in the mapreduce framework using locality sensitive hashing," International Journal of Distributed Systems and Technologies (IJDST), vol. 10, no. 4, pp. 1–16, 2019.

[20] G. Wu, Z. Zhao, G. Fu, H. Wang, Y. Wang, Z. Wang, J. Hou, and L. Huang, "A fast knn-based approach for time sensitive anomaly detection over data streams," in International Conference on Computational Science. Springer, 2019, pp. 59–74.

[21] J. Gao, H. V. Jagadish, W. Lu, and B. C. Ooi, "Dsh: data sensitive hashing for high-dimensional k-nnsearch," in Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014, pp. 1127–1138.

[22] R. F. Sproull, "Refinements to nearest-neighbor searching inkdimensional trees," Algorithmica, vol. 6, no. 1-6, pp. 579–589, 1991.

[23] K. Zhou, Q. Hou, R. Wang, and B. Guo, "Real-time kdtree construction on graphics hardware," ACM Trans. Graph., vol. 27, no. 5, pp. 126:1–126:11, Dec. 2008. [Online]. Available: http://doi.acm.org/10.1145/1409060.1409079

[24] S. M. Omohundro, Five balltree construction algorithms. International Computer Science Institute Berkeley, 1989.
[25] T. Liu, A. W. Moore, and A. Gray, "New algorithms for efficient high-dimensional nonparametric classification," Journal of Machine Learning Research, vol. 7, no. Jun, pp. 1135–1158, 2006.
[26] A. Elmasry, T. Hagerup, and F. Kammer, "Space-efficient basic graph algorithms," 2015.
[27] R. Schaffer and R. Sedgewick, "The analysis of heapsort," Journal of Algorithms, vol. 15, no. 1, pp. 76–100, 1993.
[28] A. Hinrichs, E. Novak, and H. Wozniakowski, "The curse of˝ dimensionality for monotone and convex functions of many variables," arXiv preprint arXiv:1011.3680, 2010.
[29] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[30] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist, vol. 2, 2010.
[31] Data tells various chemical combination of redwine. https:// www.kaggle.com/sh6147782/winequalityred. Accessed: 2020-1-29.
[32] Russian demography data (1990-2017). https://www.kaggle.com/dwdkills/russian-demography. Accessed: 2020-1-29.
[33] Predicting a pulsar star. https://www.kaggle.com/pavanraj159/ predicting-a-pulsar-star. Accessed: 2020-1-29.
[34] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

AUTHORS

Jude Tchaye-Kondi: Received the BS degree from the Department of Computer Science, Catholic University of West Africa. He joins in 2017 Beijing Institute of Technology, China as a graduate student and is now pursuing a Ph.D. program. His research interest includes parallel and distributed computing, edge computing, machine learning. Currently, his research projects focus on applying artificial intelligence algorithms in the edge computing environment.

Yanlong Zhai: Received the B.Eng. degree and Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004 and 2010. He is an Assistant Professor in the School of Computer Science, Beijing Institute of Technology. He was a Visiting Scholar in the Department of Electrical Engineering and Computer Science, University of California, Irvine. His research interests include cloud computing and big data.

Liehuang Zhu: Received the B.Eng. and Master Degrees in computer application from Wuhan University, Wuhan, Hubei, China, in 1998 and 2001 respectively. He received the Ph.D. degree in computer application from Beijing Institute of Technology, Beijing, China, in 2004. He is currently a Professor in the Department of Computer Science, Beijing Institute of Technology, Beijing, China. He is selected into the Program for New Century Excellent Talents in University from Ministry of Education, China. His research interests include internet of things, cloud computing security, internet, and mobile security.

FENIX: A SEMANTIC SEARCH ENGINE BASED ON AN ONTOLOGY AND A MODEL TRAINED WITH MACHINE LEARNING TO SUPPORT RESEARCH

Felipe Cujar-Rosero, David Santiago Pinchao Ortiz, Silvio Ricardo Timaran Pereira and Jimmy Mateo Guerrero Restrepo

Systems Department, University of Nariño, Pasto, Colombia

ABSTRACT

This paper presents the final results of the research project that aimed to build a Semantic Search Engine that uses an Ontology and a model trained with Machine Learning to support the semantic search of research projects of the System of Research from the University of Nariño. For the construction of FENIX, as this Engine is called, it was used a methodology that includes the stages: appropriation of knowledge, installation and configuration of tools, libraries and technologies, collection, extraction and preparation of research projects, design and development of the Semantic Search Engine. The main results of the work were three: a) the complete construction of the Ontology with classes, object properties (predicates), data properties (attributes) and individuals (instances) in Protegé, SPARQL queries with Apache Jena Fuseki and the respective coding with Owlready2 using Jupyter Notebook with Python within the virtual environment of anaconda; b) the successful training of the model for which Machine Learning algorithms and specifically Natural Language Processing algorithms were used such as: SpaCy, NLTK, Word2vec and Doc2vec, this was also done in Jupyter Notebook with Python within the virtual environment of anaconda and with Elasticsearch; and c) the creation of FENIX managing and unifying the queries for the Ontology and for the Machine Learning model. The tests showed that FENIX was successful in all the searches that were carried out because its results were satisfactory.

KEYWORDS

Search Engine, Semantic Web, Ontology, Machine Learning, Natural Language Processing.

1. INTRODUCTION

The Internet was conceived by Tim Berners-Lee as a project to manage and share knowledge and information among a select group of scientists. With the pass of the time and with the advances in the development of hardware that made possible the communication around the world, the necessary applications were developed to meet the needs of users. The large volume of content available online makes searching and processing difficult, the need to devise new ways to optimize the treatment given to such content has been vital; for the information available on the Web to be interpreted by computers without the need for human intervention, the Semantic Web is required. It is said that in Internet computers are not only capable of presenting the information contained in web pages, else they should also "understand" such information [1].

According to Berners Lee and Hendler, on the Semantic Web, information is offered with a well-defined meaning, allowing computers and people to work cooperatively. The idea behind the

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 97-115, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110709

Semantic Web is to have data on the Web defined and linked so these can be used more effectively for discovery, automatization, integration and reuse between different applications. The challenge of the Semantic Web is to offer the language that expresses data and rules to reason about many data and also allows the rules on any knowledge representation system to be exported to the Web, providing a significant degree of flexibility and "freshness" to traditional centralized knowledge representation systems, which become extremely overwhelming, and its growing in size is unmanageable. Different web systems can use different identifiers for the same concept; thus, a program that wants to compare or combine information between such systems has to know which terms mean the same thing; ideally the program should have a way of discovering the common meanings of whatever database it encounters. One solution to this problem is to add a new element to the Semantic Web; collections of information called Ontologies [2].

In the same way, it is known that the large amount of textual information available on the WEB with the increase in demand by users, makes necessary to have systems that allow access to that interest information in an efficient and effective way for saving time in the search and consultation. Among the existing techniques to achieve this efficiency and effectiveness, and in turn to provide access or facilitate the management of text document information are Machine Learning techniques, using them is highly convenient, this can be evidenced in a large number of applications in different areas [3].

This is because the factors that have generated the success of the Internet have also caused problems such as: information overload, heterogeneity of sources and consequent problems of interoperability. The Semantic Web helps to solve these problems by allowing users to delegate tasks to software tools. By incorporating semantics in the Web, the software is capable of processing content, reasoning with it, combining it and making logical deductions to solve problems automatically. Automatic ability is the result of the application of artificial intelligence techniques, which require the participation of intelligent agents that improve searches, adding values for reasoning and making decisions to web services that store high content [4].

According to Kappel, it is pertinent to make use of semantics, which is reflected in the responses that a user receives to their requests in search engines, since these go beyond the state in which users simply asked a question and received a set sorted by web page priority. Users want targeted answers to their questions without superfluous information. Answers should contain information from authorized sources, terms with the same meaning as those used in the question, relevant links, etc. So, the Semantic Web tries to provide a semantic structure to the significant contents of the Web, creating an environment in which software agents navigate through the pages performing complex tasks for users [1].

It is assumed that this Web has the ability to build a knowledge base on the preferences of users and that, through a combination of its ability to understand patterns and the information available on the Internet, it is able to meet exactly the information demands from users, for example: restaurant reservation, flight scheduling, medical consultations, purchase of books, etc. Thus, the user would obtain exact results on a search, without major complications because the Semantic Web provides a way to reason on the Web as it is an infrastructure based on metadata (highly structured data describing information), thus extending its capabilities. That is, it is not a magic artificial intelligence that allows web servers to understand the words of the users, it is only the construction of a skill arranged in a machine, in order to solve well-defined problems, through similar operations well defined to be carried out on existing data [4].

In the systematic review of the literature, a search engine is defined as an application and / or computer resource that allows information to be located on the servers of a certain website,

resulting in a list that is consistent with the files or materials stored on the corresponding servers and responding to the needs of the user. Search engines make easy to locate the information that is scattered around the Web, but it is crucial to know the way in which the search is being carried out [5]. Syntactic search engines make use of keywords, where the search result depends on an indexing process, which is the one that will allow organizing searches with these keywords or through the use of hierarchical trees categorized by a certain topic. Despite the power shown by syntactic search engines, they are still far from being able to provide to the user adequate results for the queries made, since the number of results can be too many and therefore it will be quite tedious to find the desired result or else not getting any results, with the addition that much of the responsibility for the search can fall into the hands of the user, who would have to filter and categorize their search to get a clear and concise answer [6].

In this way, it can be observed that these problems can be solved with the use of semantic search engines which, on the other hand, facilitate the user's work, are efficient in the search since they find results based on the context, thus providing information more exact about what is sought, offering a more biased number of results, facilitating the work of filtering the results by the user. In this way that these search engines interpret user searches by making use of algorithms that symbolize comprehension or understanding, offering precise results quickly and thus recognizing the correct context for the search words or sentences. It is nothing more than a semantic search engine, one that performs the search by looking at the meaning of the group of words that are written [7].

ERCIM digital library [8], NDLTD [9], Wolfram Alpha [10] use semantics to find results based on context. The last one is capable of directly answering the questions asked by the user instead of providing a list of documents or web pages that could contain the answer, as Google does. Once the question is asked, the tool calculates different answers by selectively choosing the information from the Web to end up giving a precise answer. Swotti is another search engine that uses Semantic Web technologies to extract the opinions made by users in blogs and forums about companies or products. It is able to identify the adjectives and verbs that define what people are looking for, and therefore allows people to deduce if the comment is positive or negative. When people make a search in Swotti they get not only results, else a qualitative assessment [11]. Swoogle is a document search engine for the Semantic Web, a Google for the Semantic Web although it is not aimed at the end user yet, it has been created at the University of Maryland, it is not intended for the common user, but for the crawling of semantic web documents whose formats are OWL, RDF or DAML. Swoogle is a search engine that detects, analyzes and indexes the knowledge encoded as Semantic Web documents. Swoogle understands by Semantic Web documents those that are written with some of the languages oriented to the construction of Ontologies (RDF, OWL, DAML, N3, etc). It retrieves both documents written entirely in these languages (which for Swoogle are strict Semantic Web documents) and documents partially written with some of them. It also provides an algorithm also inspired by Google's Page Rank algorithm, which for Swoogle has been called Ontology Rank. The Ontology Rank algorithm has been adapted to the semantics and usage patterns found in the Semantic Web documents. Swoogle currently has around 1.5M Semantic Web documents indexed. This information is available through an internal link to statistical data related to their status [12]. Other works such as that of Camacho Rodríguez in her undergraduate work to obtain the degree in Telematics Engineering propose incorporating a semantic search engine in the LdShake platform for the selection of educational patterns. This work was developed at the Pompeu Fabra-UPF University of Barcelona, Spain in 2013. This work analyzes the efficiency of using Ontologies to considerably improve the results and at the same time gain speed in the search [13]. Amaral presents a semantic search engine for the Portuguese language where it makes use of Natural Language Processing tools and a multilingual lexical corpus where the user's queries are evaluated, for the disambiguation of polysemic words, it uses pivots shown on the screen with the

different meanings of the word where the user chooses the meaning with which he wants to make the query [14]. Aucapiña and Plaza in their thesis for obtaining the Degree in Systems Engineering propose a semantic search engine for the University of Cuenca in Cuenca, Ecuador in 2018, where they describe in detail the use of SPARQL as a query language and the various stages carried out to achieve the prototype of the semantic search engine following proven methodologies and in certain cases those are supported by automated processes [15]. Umpiérrez Rodríguez in his final degree project in Computer Engineering called "SPARQL Interpreter" at the University of Las Palmas of Gran Canaria, developed in 2014, where he explains how SPARQL Interpreter addresses the problem of communication between a query language and a database of specific data [16]. Baculima and Cajamarca in their degree thesis in Systems Engineering developed a "Design and Implementation of an Ecuadorian Repository of Linked Geospatial Data" at the University of Cuenca Ecuador, in 2014, they work on the solution for generation, publication and visualization of data Geospatial Links, for which they rely on web search engines, this since the Web focuses on the publication of this type of data, allowing them to be structured in such a way that they can be interconnected between different sources. This work is supported by SPARQL and GEOSPARQL to be able to carry out queries, insert modification and elimination of data [17]. Iglesias, developed his project at the Simón Bolívar University of Barranquilla, his objective was to build an ontological search engine that allows semantic searches to be carried out online for master's and doctorate training works, where people can find this kind of work or topics that can serve as a guide for new research to emerge, thus improving searches when selecting research topics for undergraduate projects [18]. Bustos Quiroga in the thesis in the Master's Degree in Computer and Systems Engineering develops a "Prototype of a system for integrating scientific resources, designed to function in the space of linked open data to improve collaboration, efficiency and promote innovation in Colombia" in 2015 at the National University of Colombia. In this work he used the Semantic Web in linked data to improve integration in timelessness between applications and facilitate access to information through unified models and shared data formats [19]. Moreno and Sánchez in their undergraduate work to obtain the title of Systems and Computing Engineer propose a prototype of semantic search engines applied to the search for books on Systems Engineering and Computing in the Jorge Roa Martínez library of the Technological University of Pereira. This work was developed in 2012. This prototype was developed based on the existing theoretical foundations and the analysis that was carried out on the technologies involved, such as intelligent software agents, Ontologies that are implemented in languages such as RDF and XML, and other development tools [20]. Likewise, at the University of Nariño, Benavides and Guerrero developed the undergraduate work project to obtain the title of Systems Engineer, in 2013, called "UMAYUX: a knowledge management software model supported by a coupled-weakly dynamic Ontology with a database manager for the University of Nariño" whose objective was to convert the knowledge that was tacit, in the academic and administrative processes of the University of Nariño, into explicit knowledge that allows to collect, structure, store information and transform through the use of domain-specific Ontologies, in a way that each academic unit or administrative unit can build and couple to the model. The UMAYUX model was implemented through the construction of MASKANA, a knowledge management tool supported by a dynamic Ontology on degree works of undergraduate students of the Systems Engineering program of the Systems department of the Faculty of Engineering, weakly coupled with the PostgreSQL DBMS (Data Base Management System) [21].

Currently, the Research System of the University of Nariño does not have a tool that allows teachers, students and other researchers to carry out effective searches and queries about the research projects that have been carried out in that University. For this reason, in order to solve this problem, it was proposed to build a search engine making use of semantics through the SPARQL query language, the RDF language with the management of Ontologies and Machine Learning with a specific area called Natural Language Processing. In this way, the work can be

facilitated and the researchers and the community in general can recover and find the information requested, successfully, from the research projects that are digitized in the Research System of the University of Nariño. 85% of research projects are in Spanish language.

2. METHODOLOGY

The methodology used for the work comprises the following stages: appropriation of knowledge; installation and configuration of tools, libraries and technologies; collection, extraction and preparation of research projects; design and development of the semantic search engine.

3. RESULTS

3.1. Appropriation of knowledge.

It is highlighted the result of the acquired knowledge of all the topics covered by the project, as well as the various tools and languages used. The learning of topics such as: semantics, Semantic Web, Ontologies, Search Engines, Machine Learning, Natural Language Processing and Methontology was obtained. In the same way, the learning in languages such as Python, XML, RDF, OWL and SPARQL was known and reinforced.

3.2. Installation and configuration of tools, libraries and technologies.

It is highlighted the result of the installation and configuration of: Jupyter notebook, Protégé, Owlready2, Apache Jena Fuseki, Elasticsearch, Visual Studio Code, Anaconda, Gensim with Word2Vec and Doc2Vec, Pandas, Numpy, NLTK, SpaCy, etc.

3.3. Collection and extraction of research projects.

It is highlighted the result of collecting and extracting information from the research projects of teaching projects, student projects and degree works that are stored in the research system of the University of Nariño.

It is clarified that currently the difference between student projects and degree works is that student projects are registered from the first semesters of the university career (from first to eighth) while degree projects are registered from the last semesters of the university career (seventh onwards) until the moment of appearing as a graduate (if it is the case).

3.4. Preparation of research projects.

The result of preparing the research projects is highlighted, in such a way that this allowed for navigating through the following stages, anticipating and avoiding inconveniences, errors or problems with respect to the quality of the data.

In this order of ideas, the following phases (from the stage of preparation of research projects) are highlighted:

3.4.1. Data Organization Phase

In this phase, algorithms (created by the authors of this work) were applied to the research projects, this because the projects in the collection and extraction phase were untidy and in

conditions not suitable to be treated, managed and worked. Jupyter Notebook was used with Python and Pandas scripts to facilitate the handling of data in series and data frames.

3.4.2. Corpus Creation Phase

In this phase, the corpus for the research projects was created, which was the most powerful input of semantics, as can be seen in the later stages. This corpus resulted from unifying all the data from the research projects (already organized in the previous phase), which were: title and summary of the research; keyword 1, keyword 2, keyword 3, keyword 4, keyword 5; names, surnames, program, faculty, department, research group and line of research for each of the authors and advisers. In this phase, like the previous one, Jupyter Notebook, Python and Pandas were also used to facilitate the handling of data in series and data frames.

3.4.3. Data Pre-processing Phase

In this phase, the NLTK and SpaCy libraries were used to preprocess the data obtained in the previous phase. For this, the following subphases (from the data-preprocessing phase) were used:

3.4.3.1. Data Tokenization Subphase

In this subphase, algorithms from the NLTK library were executed to separate all the words and to be able to work with them individually.

3.4.3.2. Data Normalization Subphase

For this subphase, many algorithms were applied so that all the data were under the same standard.

3.4.3.3. Data Cleaning Subphase

In this subphase, NLTK and SpaCy algorithms were applied together with regular expressions so that the data is totally clean, this with the elimination of null data, punctuation marks, “non-ascii” characters and stopwords.

3.4.3.4. Data Lemmatization Subphase

Finally, in this subphase, the data resulting from the cleaning stage were lemmatized.

3.5. Design of Semantic Search Engine

Once the previous stage of preparation of the research projects was completed, FENIX was designed. This design was carried out taking into account the specification and conceptualization phases of the Methontology methodology, where the following results stand out:

3.5.1. Specification Phase

Within this phase, the reasons that that allowed to make the Ontology were identified, it was also described the end users who make use of the Semantic Search Engine. Also a knowledge acquisition process was carried out; this process of acquiring knowledge differs from the appropriation of the general knowledge of the project, since this acquisition is mainly focused on the design of the Search Engine.

Thus, it is highlighted that the Ontology was created to give a robust semantic component to the Search Engine interacting with the research projects and their components. The end users will be all those researchers who wish to access the research project resources through various searches.

3.5.2. Conceptualization Phase

Within the conceptualization phase, eleven specific tasks were developed that allowed to successfully conceptualize: classes, attributes, relationships and instances of Ontology. These tasks were:

Task 1. Build the glossary of terms:

This task listed all the important terms selected after analyzing the previous specification phase with its knowledge acquisition process, also this task presented a brief description of each term as shown in Table 1.

Table 1. Glossary of terms of Ontology

Term	Description
Universidad	The education-oriented entity that contains faculties.
Facultad	The entity that contains academic departments.
VIIS	Vice-Chancellor of Research and Social Interaction, is the entity in charge of the research aspect throughout the University, is the one who manages the economic resources for research projects.
Departamento	The entity that contains academic programs.
Convocatoria	This term refers to the convocatory by the VIIS for researchers to come to this convocatory and submit projects in order for them to be financed.
Programa	It is the academic program that is conformed by teachers and students.
Grupo de investigación	It is the group conformed by teachers and/or research students in order to submit projects to the VIIS convocatory.
Docente	He/She is a researcher who belongs to the University, who carries out projects of teaching type.
Estudiante	He/She is a researcher who belongs to the University, who carries out projects of student type and/or degree works.
Investigador externo	He/She is a researcher who is external to the University but who presents for the convocatory for VIIS.
Línea de investigación	It is a branch that the research group manages, focused on a specific area of knowledge.
Investigador	He/She is the one who develops research projects and submits them to the VIIS convocatory. This researcher may be a teacher, student or external researcher.
Proyecto de investigación	It is perhaps the most important entity within the research domain that contains everything related to a research project.
Palabra	This entity refers to each of the words that conforms the research project, these were used for building the thesaurus and generating a big part of the semantic.

Task 2. Build concept taxonomies:

This task defined the taxonomy or hierarchy of ontology concepts or classes that were obtained from the glossary of terms in task 1, this taxonomy is shown in Figure. 1.

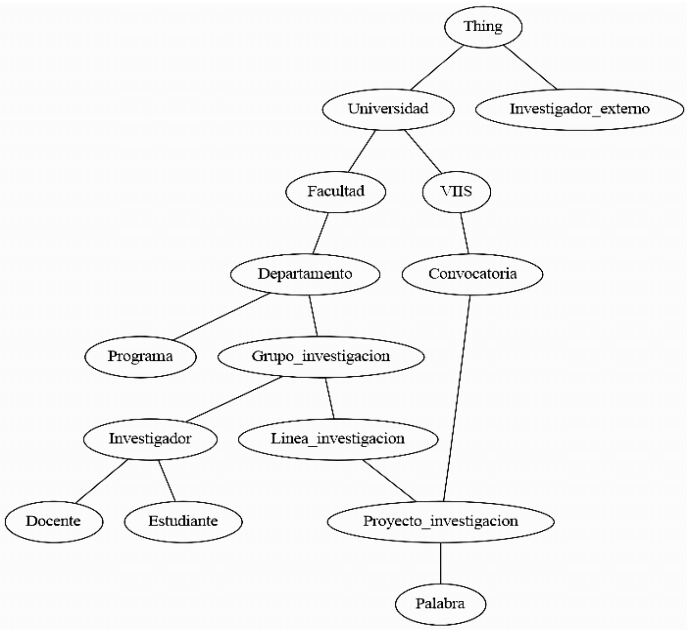


Figure. 1 Taxonomy of ontology concepts

Task 3. Build ad hoc binary relation diagrams:

In this task the binary relations diagram that contains the predicates of Ontology was elaborated. The relations of the most important class of Ontology are visualized in Figure. 2, which is "Proyecto de investigación".

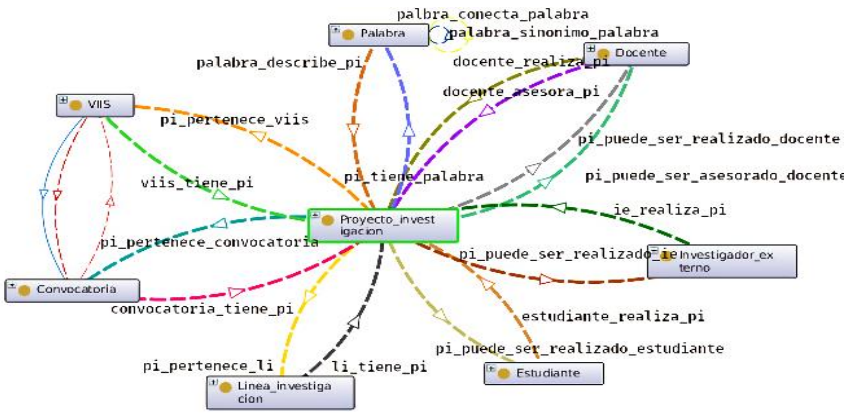


Figure. 2 Binary relations diagram for class: "Proyecto de Investigación"

Task 4. Build concepts dictionary:

This task detailed the most important or relevant concepts or classes within the research domain of Ontology, highlighting its attributes, relations and instances. "Proyecto de investigación" was

chosen because this class have all the raw material with many words for discovering the semantic power. It is shown in Table 2.

Table 2. Concept dictionary for "Proyecto de Investigación"

Class Proyecto de Investigación	
Attributes	id_proyecto_investigacion titulo_proyecto_investigacion resumen_proyecto_investigacion palabras_clave estado_proyecto_investigacion tipo_proyecto_investigacion
Relations	pi_tiene_palabra pi_pertenece_li pi_pertenece_convocatoria pi_puede_ser_realizado_estudiante pi_puede_ser_realizado_docente pi_puede_ser_realizado_ie pi_puede_ser_asesorado_docente pi_pertenece_viis
Instances	Proyecto investigación

Task 5. Describe ad hoc binary relations:

A total of 55 binary relationships were obtained, of which "pi_tiene_palabra" is highlighted because each project relates to its words in such "pi_tiene_palabra" relation facilitates searches.

It is shown in Table 3.

For each relationship its origin class (domain), destination class (range), inverse relation and cardinality were obtained.

Table 3. Binary relation (pi_tiene_palabra) in detail

Relation pi_tiene_palabra	
Origin Class (Domain)	Proyecto de investigación
Destination Class (Range)	Palabra
Inverse Relation	palabra_describe_pi
Cardinality	1:N

Task 6. Describe instance attributes:

A total of 45 attributes were obtained, of which the most representative class ("Proyecto de investigación") is shown. Instance attributes of this class can be seen in Table 4.

For each class, the instance attributes, the class name (domain), the data type (range) and its cardinality are displayed.

Table 4. Instance attributes for class: “Proyecto de Investigación”

Attribute	Data Type (Range)	Cardinality
id_proyecto_investigacion	Int	1
titulo_proyecto_investigacion	String	1
resumen_proyecto_investigacion	String	1
palabras_clave	String	0:5
estado_proyecto_investigacion	String	1
tipo_proyecto_investigacion	String	1

Task 7. Describe class attributtes:

This task defined class attributes that serve as cardinal constraints for each class. These can be observed in Table 5 for classes “Investigador”, “Docente” and “Proyecto de Investigación”.

Table 5. Class attributes for: “Investigador”, “Docente” and “Proyecto de Investigación”

Class	Attribute
Investigador	Maximum 4 researchers per research project.
Docente	Maximum 2 teachers can advise research projects.
Proyecto de Investigación	Maximum 2 years and minimum 6 months duration of the research project.

Task 8. Describe constants:

The need to use constants for this Ontology was not contemplated.

Task 9. Describe formal axioms:

There was no need to use axioms that are predicates (relationships) that are always fulfilled, i.e. they are always affirmative.

Task 10. Describe rules:

Because this Ontology did not introduce formal axioms, no rules were necessary.

Task 11. Describe instances:

Instances were obtained for each of the classes contemplated in ontology: “Universidad, Facultad, VIIS, Departamento, Convocatoria, Programa, Grupo de investigación, Docente, Estudiante, Investigador externo, Línea de investigación, Investigador, Proyecto de investigación y Palabra”. This was achieved with the previous stage of preparation of research projects.

3.6. Development of the Semantic Search Engine

FENIX was developed based on three phases (from the stage of Development of the Semantic Search Engine) in which the following results are highlighted:

3.6.1. Development with Methontology Phase

For this phase the three subphases of Methontology were applied which are: formalization, implementation and evaluation.

3.6.1.1. Formalization Subphase

This phase highlights the results obtained after using the Protégé tool for the construction of Ontology in semi-computable terms.

3.6.1.2. Implementation Subphase

This phase highlights the results of using the Owlready2 library to encode a computable version of Ontology. Scripts were created and encoding was performed for the handling of Ontology with Python where an entire process of instantiating objects of all classes was performed:

Owlready2 “DataProperties” that correspond to ontology attributes, along with Owlready2 “ObjectProperties” that correspond to Ontology relations were also encoded within the scripts; for each of these elements mentioned, the domain and range were determined. It should be said that Owlready2 reverse relationships are executed in the background, so it was only enough execute the direct relationship.

In synthesis, all classes, attributes, and relations were instantiated within Ontology.

3.6.1.3. Evaluation Subphase

This phase highlights results after having performed functional tests locally and having successfully retrieved the data and other components of ontology with the use of SPARQL and Apache Jena Fuseki server by handling triples of RDF (subject predicate object).

3.6.2. Development with Machine Learning Phase

This phase highlights the results of training with the Machine Learning algorithm with Natural Language Processing such as Word2Vec, which helped to find the context that a word has, in addition a model was trained with the Doc2Vec algorithm, which relies on Word2Vec to find documents that relate to each other, these models make use of neural networks. In this case, the model was trained with the algorithms previously mentioned based on the Skip-Gram model, which attempts to predict words or documents in context given a word or set of base words to search for.

It should be clarified that the output returned by Word2Vec was the input for the process performed with Doc2Vec, this is possible since both algorithms work hand in hand to achieve discover semantic relationships and retrieve information semantically effectively.

To perform the search for similarity between words or documents, of a set of given words, the Gensim library was used, which makes use of the normalization of the vectors obtained from the words to be searched and the calculation of the product point between the normalized vector and each of the vectors corresponding to each word or document trained.

The model was created with data from the preparation stage of research projects, the respective hyper parameters were assigned, the model was trained, the results were evaluated and the hyper parameters were re-fed to satisfactory results, as it is evidenced in Table 6.

Table 6. Hyperparameters for word2vec and doc2vec models

Name	Value	Description
vector_size	300	Dimension of the vector of each of the words in the corpus.
window	5	Refers to the context where the distance between predicted words is chosen.
min_count	1	Minimum words to look for.
dm	0	0 indicates that Doc2Vec PV-DBOW is used which is analogous to the Skip-Gram model used in Word2vec. 1 indicates that Doc2Vec PV-DM is used which is analogous to the CBOW model used in Word2Vec.
dbow_words	1	0 indicates that it will train with Doc2Vec. 1 indicates that it will train with Doc2Vec taking Word2Vec input.
hs	0	It is the value with which the neuron will be punished in case the task done is not correct.
negative	20	Number of irrelevant words for negative sampling.
ns_exponent	-0.5	Indicates that frequencies will be sampled equally.
alpha	0.015	Neural network learning rate
min_alpha	0.0001	Rate to be reduced during training.
seed	25	Seed to generate hash for words.
sample	5	Reduction number for high frequency words
epochs	150	Epochs, number of iterations for training.

In Figure. 3, Figure. 4 and Figure. 5 are presented the results of executing the order to find 10 more similar and related words (according to the cosine similarity of the algorithm ordered in percentage terms from highest to lowest) to another word that is specified within of the entire research corpus with a method of the Word2Vec algorithm.

Figure. 3 indicates the 10 words most similar and related to the word “cultivos”.

```
modelo_cargado.wv.most_similar(
    positive=['cultivos'], topn=10)
```

```
[('andinos', 0.5301966667175293),
 ('sustituir', 0.5119062662124634),
 ('agrotecnologias', 0.4994411766529083),
 ('totipotencia', 0.49643680453300476),
 ('cebada', 0.49597498774528503),
 ('invitro', 0.49116745591163635),
 ('transitorios', 0.4897231459617615),
 ('hechas', 0.48805299401283264),
 ('potencializador', 0.4799562096595764),
 ('recesivo', 0.47675687074661255)]
```

Figure. 3 Result of method with Word2vec for word “cultivos”

Figure. 4 indicates the 10 words most similar and related to the word “fresa”.

```
modelo_cargado.wv.most_similar(
    positive=['fresa'], topn=10)
```

```
[('cereza', 0.9345278739929199),
 ('citricos', 0.9311402440071106),
 ('ciruela', 0.9139297604560852),
 ('pera', 0.8887712359428406),
 ('yogurt', 0.7925349473953247),
 ('banano', 0.7829478979110718),
 ('manzana', 0.7650518417358398),
 ('subtropico', 0.6739339828491211),
 ('canada', 0.6399896144866943),
 ('usado', 0.6375434994697571)]
```

Figure. 4 Result of method with Word2vec for word “fresa”

Figure. 5 indicates the 10 words most similar and related to the word “historia”.

```
modelo_cargado.wv.most_similar(
    positive=['historia'], topn=10)
```

```
[('guerreras', 0.5881358981132507),
 ('feminista', 0.5563334822654724),
 ('musicologia', 0.5516680479049683),
 ('diversion', 0.5437859296798706),
 ('empoderarse', 0.5405762195587158),
 ('juventud', 0.5386302471160889),
 ('constatado', 0.5351422429084778),
 ('enhem', 0.5351074934005737),
 ('amor', 0.5341321229934692),
 ('resignificacion', 0.53216552734375)]
```

Figure. 5 Result of method with Word2vec for word “historia”

3.6.3. Integration of Ontology and Machine Learning Phase

For this phase, Ontology and Machine learning are integrated, providing potency, effectiveness and semantic power to optimize times, resources, and to have greater chances of finding successful and satisfactory results to certain searches in FENIX, the results are observed in: Figure. 6, Figure. 7 and Figure. 8.

This was achieved by bringing the vectors that Doc2Vec generated to Elasticsearch; Elastic helped in the ranking stage by having speed, scalability and being a distributed analysis engine that favors the search and indexing of research projects.

Afterwards, scripts were created to manage the queries of the research projects for the Ontology with SPARQL, which relies on the trained Word2Vec model to add additional words to the search that are related to those requested and thus find research related to a certain query. In the same way, with Doc2Vec it was possible to infer vectors from a set of supplied words, then as a partial result, the investigations that are related to inferred vectors are presented. Finally, the results obtained in the SPARQL query and the Doc2Vec algorithm are joined, so the final ranking of a search will show consistent, coherent, successful and satisfactory results as requested with the additional ability to recommend documents that may be useful and interest to the user.

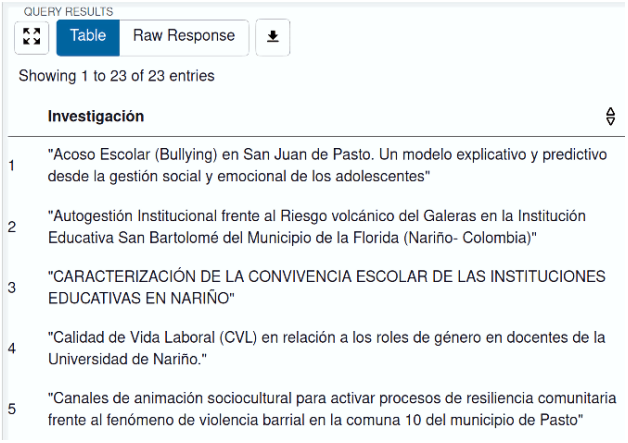

QUERY RESULTS

Table | Raw Response

Showing 1 to 23 of 23 entries

	Investigación
1	"Acoso Escolar (Bullying) en San Juan de Pasto. Un modelo explicativo y predictivo desde la gestión social y emocional de los adolescentes"
2	"Autogestión Institucional frente al Riesgo volcánico del Galeras en la Institución Educativa San Bartolomé del Municipio de la Florida (Nariño- Colombia)"
3	"CARACTERIZACIÓN DE LA CONVIVENCIA ESCOLAR DE LAS INSTITUCIONES EDUCATIVAS EN NARIÑO"
4	"Calidad de Vida Laboral (CVL) en relación a los roles de género en docentes de la Universidad de Nariño."
5	"Canales de animación sociocultural para activar procesos de resiliencia comunitaria frente al fenómeno de violencia barrial en la comuna 10 del municipio de Pasto"

Figure. 6 First results for the search: “investigaciones de psicología”

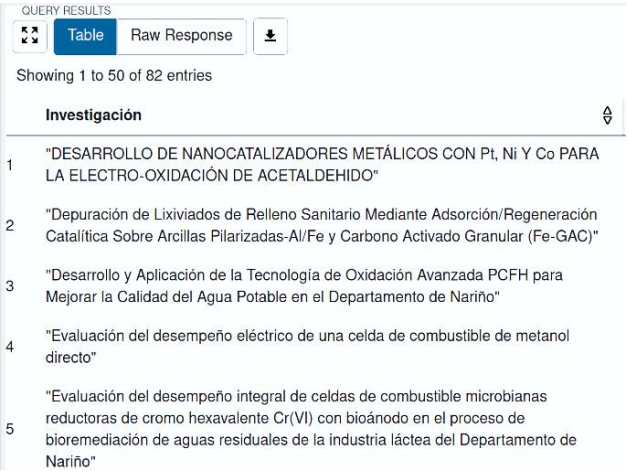

QUERY RESULTS

Table | Raw Response

Showing 1 to 50 of 82 entries

	Investigación
1	"DESARROLLO DE NANOCATALIZADORES METÁLICOS CON Pt, Ni Y Co PARA LA ELECTRO-OXIDACIÓN DE ACETALDEHIDO"
2	"Depuración de Lixiviados de Relleno Sanitario Mediante Adsorción/Regeneración Catalítica Sobre Arcillas Pilarizadas-Al/Fe y Carbono Activado Granular (Fe-GAC)"
3	"Desarrollo y Aplicación de la Tecnología de Oxidación Avanzada PCFH para Mejorar la Calidad del Agua Potable en el Departamento de Nariño"
4	"Evaluación del desempeño eléctrico de una celda de combustible de metanol directo"
5	"Evaluación del desempeño integral de celdas de combustible microbianas reductoras de cromo hexavalente Cr(VI) con bioánodo en el proceso de bioremediación de aguas residuales de la industria láctea del Departamento de Nariño"

Figure. 7 First results for the search: “investigaciones sobre química”

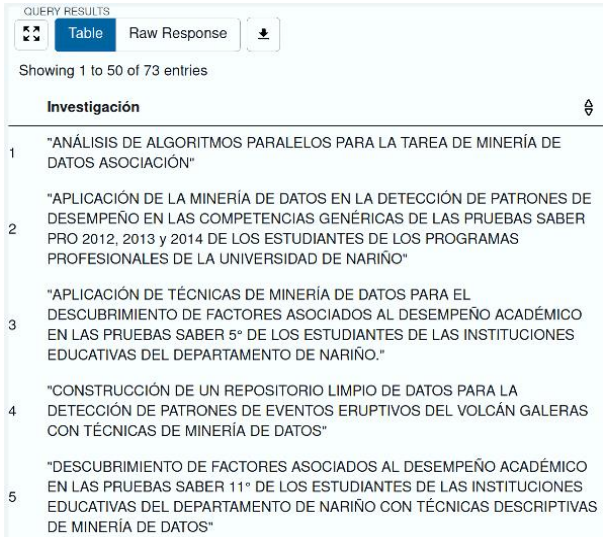

QUERY RESULTS

Table | Raw Response

Showing 1 to 50 of 73 entries

	Investigación
1	"ANÁLISIS DE ALGORITMOS PARALELOS PARA LA TAREA DE MINERÍA DE DATOS ASOCIACIÓN"
2	"APLICACIÓN DE LA MINERÍA DE DATOS EN LA DETECCIÓN DE PATRONES DE DESEMPEÑO EN LAS COMPETENCIAS GENÉRICAS DE LAS PRUEBAS SABER PRO 2012, 2013 y 2014 DE LOS ESTUDIANTES DE LOS PROGRAMAS PROFESIONALES DE LA UNIVERSIDAD DE NARIÑO"
3	"APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS PARA EL DESCUBRIMIENTO DE FACTORES ASOCIADOS AL DESEMPEÑO ACADÉMICO EN LAS PRUEBAS SABER 5° DE LOS ESTUDIANTES DE LAS INSTITUCIONES EDUCATIVAS DEL DEPARTAMENTO DE NARIÑO."
4	"CONSTRUCCIÓN DE UN REPOSITORIO LIMPIO DE DATOS PARA LA DETECCIÓN DE PATRONES DE EVENTOS ERUPTIVOS DEL VOLCÁN GALERAS CON TÉCNICAS DE MINERÍA DE DATOS"
5	"DESCUBRIMIENTO DE FACTORES ASOCIADOS AL DESEMPEÑO ACADÉMICO EN LAS PRUEBAS SABER 11° DE LOS ESTUDIANTES DE LAS INSTITUCIONES EDUCATIVAS DEL DEPARTAMENTO DE NARIÑO CON TÉCNICAS DESCRIPTIVAS DE MINERÍA DE DATOS"

Figure. 8 First results for the search: “investigaciones acerca de minería de datos”

4. DISCUSSION

√ FENIX provides a degree of optimization, originality and innovation compared to other search engines and knowledge databases such as WordNet, Freebase, DLBP (Digital Bibliography and Library Project), ERCIM digital library, Swoogle, NDLTD, Wolfram Alpha (and others mentioned above) because ontologies are being integrated with Machine Learning with the aforementioned scripts where the ontology is well set up and the algorithms are well trained. In addition to this, the vectors of the words are being managed with Elasticsearch, which save significant memory consumption. Searches are also done with Elasticsearch which is another reason because the search engine is so fast and accurate.

√ It is recommended to carry out tests with more data to see how FENIX behaves in the face of an expandable size in the information. This is because the data used were all the research projects that were in the VIIS Research System, but all the projects at the University level are not within that system, but 10%.

√ It is proposed to do the coupling of FENIX in other universities and in various non-academic environments, determining the structure of the Ontology and Machine Learning models with their possible variants.

√ It is suggested to carry out an analysis of what users are looking for, analyzing the records of searches, downloads, storing everything in the database, then applying data mining with all the information to possibly determine aspects such as: “What semesters do belong people who make queries about astronomy ?” or “What ages do belong people who make queries about psychology?”. Machine Learning could be used for this future work perfectly.

√ It is also proposed to incorporate in the search engine page a view with its respective database that allows to rate and comment on the search engine in order to observe and analyze how users are rating FENIX, as well as to realize their opinions and whether they are satisfied or not, thus determining the usability of FENIX.

5. CONCLUSIONS

√ With the culmination of this research work, FENIX is obtained: A Semantic Search Engine based on an Ontology and a Machine Learning model for research projects at the University of Nariño. Through the successful development of the project stages, the formulated problem is solved, the objectives set are fullfiled and satisfactory results are obtained. In this way, this tool facilitates the successful search for research projects for teaching projects, student projects and degree projects at the University of Nariño.

√ In the stages of appropriation of knowledge and installation and configuration of the tools, a domain of the various topics was acquired and this contributed to the development of the work and led to the personal training of the researchers as well as made outstanding contributions to the group of GRIAS research (Grupo de Investigación Aplicado en Sistemas) and for the University of Nariño in general.

√ The stages of collection, extraction and preparation of research projects were extremely important stages that acted as preliminary and prelude stages as input for FENIX. In this vein, it is correct to affirm that without these stages a good development of FENIX could not have been achieved.

√ Methontology was a methodology that was perfectly coupled to the project and allowed to build the Ontology following specific phases and tasks with an order, comprehension and accuracy in the processes.

√ The Ontology integrated with Machine Learning demonstrated great potency, semantic power and effectiveness in the processes to obtain concrete results according to the searches carried out. This is because Machine Learning algorithms, specifically Natural Language Processing algorithms such as Word2vec and Doc2vec work with neural networks, which were trained with the words from the research project corpus, adapting them to the context and finding the various semantic relationships between them. Likewise, Ontology acted as a great semantic network whose instances, hand in hand with classes, relations and attributes, interacted under the triple scheme handled by RDF and consulted by SPARQL to extract all the knowledge from the domain of the research projects.

ACKNOWLEDGMENT

To the University of Nariño, to the VIIS (Vicerrectoría de Investigación e Interacción Social) for financing this project and to the Research Community in general for supporting the successful completion of this work.

REFERENCES

[1] VELÁSQUEZ, Torcoroma, PUENTES, Andrés & GUZMÁN, Jaime. Ontologías: una técnica de representación de conocimiento. En: Avances en Sistemas e Informática. Vol. 8. No. 2. (Julio, 2011), p. 211-216. [En línea]. Disponible en: https://revistas.unal.edu.co/index.php/avances/article/view/26750

[2] GARCÍA, Francisco. Web Semántica y Ontologías. [En línea]. Disponible en: https://www.researchgate.net/publication/267222548_Web_Semantica_y_Ontologias

[3] MOURIÑO, M. Clasificación multilingüe de documentos utilizando machine learning y la wikipedia. [En línea]. Disponible en: https://dialnet.unirioja.es/servlet/tesis?codigo=150295

[4] EFIGENIA, Ana & CANTOR, Sandoval. USO DE ONTOLOGÍAS Y WEB SEMÁNTICA PARA APOYAR LA GESTIÓN DEL CONOCIMIENTO. En: Ciencia e Ingeniería Neogranadina. Vol. 17 No. 2. (Diciembre, 2007), p.111-129. [En línea]. Disponible en: https://dialnet.unirioja.es/descarga/articulo/2512191.pdf

[5] GALLO, Manuel, FABRE, Ernesto & GALLO, Manuel. ¿Qué es un buscador? [En línea]. Disponible en: http://media.axon.es/pdf/98234_1.pdf

[6] FAZZINGA, Bettina, GIANFORME, Giorgio, GOTTLOB, Georg & LUKASIEWICZ, Thomas. Semantic Web Search Based On Ontological Conjunctive Queries. En: SSRN Electronic Journal. [En línea]. Disponible en: https://www.researchgate.net/publication/326473981_Semantic_Web_Search_Based_on_Ontological_Conjunctive_Queries

[7] DE PEDRO, A. Buscadores Semánticos, para qué sirven. Usos en la AAPP. [En línea]. Disponible en: http://www.alejandrodepedro.es/buscadores-semanticos-el-paso-al-30

[8] ANDREONI, Antonella, BALDACCI Maria, BIAGONI, Stefania, CARLESI, Carlo, CASTELLI, Donatella, PAGANO, Pasquale, PETERS, Carol & PISANI, Serena. The ERCIM Technical Reference Digital Library. En: D-Lib Magazine. Vol. 5. No. 12. (Diciembre, 1999). [En línea]. Disponible en: http://www.dlib.org/dlib/december99/peters/12peters.html

[9] NDLTD. Networked Digital Library of Theses and Dissertations. [En línea]. Disponible en: http://www.ndltd.org

[10] WolframAlpha Computational Intelligence. [En línea]. Disponible en: https://www.wolframalpha.com

[11] MARTÍN, Javier. Swotti buscador de opiniones. [En línea]. Disponible en: https://loogic.com/swotti-buscador-de-opiniones

[12] BARBERÁ, Consuelo, MILLET, Mercé & TORRES, Emiliano. Estudio del buscador semántico Swoogle. [En línea]. Disponible en: https://www.uv.es/etomar/trabajos/swoogle/swoogle.pdf

[13] CAMACHO, María. Incorporación de un buscador semántico en la plataforma LdShake para la selección de patrones educativos. Barcelona, 2013, 76p. Trabajo de grado. Universidad Pompeu Fabra. Escuela Superior Politécnica UPF. Ingeniería de Telemática. [En línea]. Disponible en: https://repositori.upf.edu/handle/10230/22172

[14] AMARAL, Carlos, LAURENT, Dominique, MARTINS, André, MENDES, Alfonso & PINTO, Cláudia. Design and Implementation of a Semantic Search Engine for Portuguese. [En línea]. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.4090&rep=rep1&type=pdf

[15] AUCAPIÑA, Yolanda & PLAZA, C. Buscador semántico universitario: Caso de estudio Universidad de Cuenca. Cuenca, 2018, 200p. Trabajo de grado (Tesis previa a la obtención del Título de Ingeniero en Sistemas). Universidad de Cuenca. Facultad de Ingeniería. Ingeniería de Sistemas. [En línea]. Disponible en: http://dspace.ucuenca.edu.ec/handle/123456789/30291

[16] UMPIÉRREZ, Francisco. SPARQL Interpreter. Las Palmas de Gran Canaria, 2014, 65p. Trabajo de grado (Trabajo Final de Grado en Ingeniería Informática). Universidad de Las Palmas de Gran Canaria. Escuela Ingeniería Informática. Ingeniería Informática. [En línea]. Disponible en: https://nanopdf.com/download/0701044000000000pdf_pdf

[17] BACULIMA, Jhon & CAJAMARCA, Marcelo. Diseño e Implementación de un Repositorio Ecuatoriano de Datos Enlazados Geoespaciales. Cuenca, 2014, 131p. Trabajo de grado (Tesis de Grado previa a la obtención del Título: Ingeniero de Sistemas). Universidad de Cuenca. Facultad de Ingeniería. Ingeniería de sistemas. [En línea]. Disponible en: http://dspace.ucuenca.edu.ec/handle/123456789/19876

[18] IGLESIAS, Daniela, MEJÍA, Omar, NIETO, Julio, SÁNCHEZ, Steven & MORENO, Silvia. Construcción de un buscador ontológico para búsquedas semánticas de proyectos de maestría y doctorado. En: Investigación y Desarrollo en TIC. Vol. 7. No. 1. (Mayo, 2017), p. 7-13. [En línea]. Disponible en: https://revistas.unisimon.edu.co/index.php/identic/article/view/2501

[19] BUSTOS, Gabriel. Prototipo de un sistema de integración de recursos científicos, diseñado para su funcionamiento en el espacio de los datos abiertos enlazados para mejorar la colaboración, la eficiencia y promover la innovación en Colombia. Bogotá, 2018. Tesis de Maestría. Universidad Nacional de Colombia. Facultad de Ingeniería. Ingeniería de Sistemas e Industrial. [En línea]. Disponible en: https://repositorio.unal.edu.co/handle/unal/55245

[20] MORENO, Carlos & SÁNCHEZ, Yakeline. Prototipo de buscador semántico aplicado a la búsqueda de libros de Ingeniería de Sistemas y Computación en la biblioteca Jorge Roa Martínez de la Universidad Tecnológica de Pereira. Pereira, 2012, 66p. Trabajo de grado. Universidad Tecnológica de Pereira. Facultad de Ingenierías: Eléctrica, Electrónica, Física y Ciencias de la Computación. Ingeniería De Sistemas y Computación. [En línea]. Disponible en: http://repositorio.utp.edu.co/dspace/bitstream/11059/2671/1/0057565M843.pdf

[21] BENAVIDES, Mauricio & GUERRERO, Jimmy. UMAYUX: un modelo de software de gestión de conocimiento soportado en una ontología dinámica débilmente acoplado con un gestor de base de datos. San Juan de Pasto, 2014, 145p. Trabajo de grado (Trabajo de grado presentado como requisito parcial para optar al título de Ingeniero de Sistemas). Universidad de Nariño. Facultad de Ingeniería. Ingeniería de Sistemas. [En línea]. Disponible en: http://sired.udenar.edu.co/2030

[22] Apache Software Foundation. Apache Jena Fuseki. [En línea]. Disponible en: https://jena.apache.org/documentation/fuseki2

[23] ARAUJO, Joaquín. ¿Qué es Docker? ¿Qué son los contenedores? y ¿Por qué no usar VMs? [En línea]. Disponible en: https://platzi.com/tutoriales/1432-docker/1484-guia-del-curso-de-docker

[24] BUDHIRAJA, Amar. A simple explanation of document embeddings generated using Doc2Vec. [En línea]. Disponible en: https://medium.com/@amarbudhiraja/understanding-document-embeddings-of-doc2vec-bfe7237a26da

[25] CHALLENGER, Ivet, DÍAZ, Yanet & BECERRA, Roberto. El lenguaje de programación Python. En: Ciencias Holguín. Vol. XX. No. 2. (Junio, 2014), p. 1-13. [En línea]. Disponible en: www.redalyc.org/articulo.oa?id=181531232001

[26] CHECA, Diego & ROJAS, Oscar. ONTOLOGÍA PARA LOS SISTEMAS HOLÓNICOS DE MANUFACTURA BASADOS EN LA UNIDAD DE PRODUCCIÓN. En: Revista Colombiana de Tecnologías de Avanzada. Vol. 1. No. 23. (Noviembre, 2013), p. 134-141. [En línea]. Disponible en: http://revistas.unipamplona.edu.co/ojs_viceinves/index.php/RCTA/article/view/2334

[27] CLASSORA. Sacando provecho a la Web Semántica: SPARQL. [En línea]. Disponible en: http://blog.classora.com/2012/11/05/sacando-provecho-a-la-web-semantica-sparql
[28] CODINA, Lluís & CRISTÒFOL, Rovira. La Web Semántica. En: Jesús Tramullas (coord.). Tendencias en documentación digital. Guijón: Trea, 2006. p. 9-54. [En línea]. Disponible en: http://eprints.rclis.org/8899
[29] EDWARDS, Gavin. Machine Learning An Introduction. [En línea]. Disponible en: https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0
Elasticsearch B.V. Elasticsearch. [En línea]. Disponible en: https://www.elastic.co/es/what-is/elasticsearch
[30] FLORES, Pedro & PORTILLO, Julio. ELABORACIÓN DE PROPUESTA DE GUÍA DE IMPLEMENTACIÓN DE SCRUM PARA EMPRESA SALVADOREÑA, UN CASO DE ESTUDIO. Antiguo Cuscatlán, 2017, 117p. Trabajo de grado (MAESTRO EN ARQUITECTURA DE SOFTWARE). Universidad Don Bosco. Arquitectura de Software. [En línea]. Disponible en: http://rd.udb.edu.sv:8080/jspui/bitstream/11715/1264/1/documento.pdf
[31] FLÓREZ, Héctor. Construcción de ontologías OWL. En: VÍNCULOS. Vol. 4. No. 1. (Diciembre, 2007), p. 19-34. [En línea]. Disponible en: https://revistas.udistrital.edu.co/index.php/vinculos/article/view/4112
[32] Kit de herramientas de lenguaje natural. [En línea]. Disponible en: https://www.nltk.org
[33] LAMY, Jean. Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. En: Artificial Intelligence in Medicine. Vol. 80. (Agosto, 2017), p. 11-28. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0933365717300271
[34] LINCOLN, Matthew. Uso de SPARQL para acceder a datos abiertos enlazados. [En línea]. Disponible en: https://programminghistorian.org/es/lecciones/sparql-datos-abiertos-enlazados
[35] LOZANO, Adolfo. Ontologías en la Web Semántica. [En línea]. Disponible en: http://eolo.cps.unizar.es/docencia/MasterUPV/Articulos/Ontologias%20en%20la%20Web%20Semantica.pdf
[36] MUÑOZ, José. Introducción a flask. [En línea]. Disponible en: https://plataforma.josedomingo.org/pledin/cursos/flask/curso/u05/
[37] PEDRAZA, Rafael, CODINA, Lluís & CRISTÒFOL, Rovira. Web semántica y ontologías en el procesamiento de la información documental. En: El profesional de la información. Vol. 16. No. 6. (Noviembre, 2007), p. 569-579. [En línea]. Disponible en: https://repositori.upf.edu/handle/10230/13141
[38] PEREZ, Fernando & GRANGER, Brian. Project Jupyter. [En línea]. Disponible en: https://jupyter.org PostgreSQL. [En línea]. Disponible en https://www.postgresql.org/about/
[39] ROCCA, Joseph. A simple introduction to Machine Learning. [En línea]. Disponible en: https://towardsdatascience.com/introduction-to-machine-learning-f41aabc55264
[40] SHETTY, Badreesh. Natural Language Processing (NLP) for Machine Learning. [En línea]. Disponible en: https://towardsdatascience.com/natural-language-processing-nlp-for-machine-learning-d44498845d5b
[41] SHPERBER, Gidi. A gentle introduction to Doc2Vec. [En línea]. Disponible en: https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e
[42] Sistema de Información de Investigaciones. [En línea]. Disponible en: http://sisinfoviis.udenar.edu.co
[43] SpaCy 101: todo lo que necesita saber. [En línea]. Disponible en: https://spacy.io/usage/spacy-101
[44] TABARES, John & JIMÉNEZ, Jovani. Ontología para el proceso evaluativo en la educación superior. En: Revista Virtual Universidad Católica del Norte. Vol. 1. No. 42. (Agosto, 2014), p. 68-79. [En línea]. Disponible en: https://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/495

AUTHORS

FELIPE CUJAR ROSERO: Research student of the GRIAS group of the University of Nariño with publication of papers, presentations, poster exhibition and certifications in the areas of database knowledge, artificial intelligence and web development. Link: https://www.linkedin.com/in/felipe-cujar/
https://scholar.google.com/citations?user=dX12cEAAAAAJ
https://scienti.minciencias.gov.co/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001853544

DAVID SANTIAGO PINCHAO ORTIZ: Research student of the GRIAS group of the University of Nariño with publication of papers, presentations, poster exhibition and certifications in the areas of database knowledge, artificial intelligence and web development. Link: https://co.linkedin.com/in/sangeeky

SILVIO RICARDO TIMARÁN PEREIRA: Doctor of Engineering. Director of Research Group GRIAS. Professor in the Systems Department of the University of Nariño. Link: Researcher: http://scienti.colciencias.gov.co:8081/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0000250988

MATEO GUERRERO RESTREPO: Master in Engineering. GRIAS Group Researcher. Professor Hora chair of Systems Department of the University of Nariño. Link: Researcher: http://scienti.minciencias.gov.co:8081/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001489230

A NATURAL LOGIC FOR ARTIFICIAL INTELLIGENCE, AND ITS RISKS AND BENEFITS

Gyula Klima

Department of Philosophy, Fordham University, New York, USA

ABSTRACT

This paper is a multidisciplinary project proposal, submitted in the hopes that it may garner enough interest to launch it with members of the AI research community along with linguists and philosophers of mind and language interested in constructing a semantics for a natural logic for AI. The paper outlines some of the major hurdles in the way of "semantics-driven" natural language processing based on standard predicate logic and sketches out the steps to be taken toward a "natural logic", a semantic system explicitly defined on a well-regimented (but indefinitely expandable) fragment of a natural language that can, therefore, be "intelligently" processed by computers, using the semantic representations of the phrases of the fragment.

KEYWORDS

Natural logic, natural vs. artificial intelligence, semantics-driven language processing.

1. INTRODUCTION

The purpose of this proposal is to launch a research project facilitating *intelligent natural language processing by computers*. In order to do so, we intend to pool the resources of scholars working in various fields, ranging from linguistics, philosophy, and history of logic, as well as model theoretical semantics and formal ontology, to computer science and artificial intelligence, cognitive psychology, and ethics. The need for such wide-ranging collaboration becomes apparent once we consider the enormity of the task.

2. BACKGROUND AND MOTIVATION

To make computers "intelligent", at least to the extent of being capable of processing bits and pieces of a human language with some semblance of understanding, we need to understand what the intelligent use of language consists in. For of course linguistic symbols can be used, processed, and manipulated without any understanding whatsoever, as it happens in simple word processors. Indeed, through the interactions of an intelligent user, even such primitive symbol manipulators can produce useful results, such as neat formatting, spell-checking, or answering simple questions about a text (or even an enormous number of documents) through string-searches and other mere syntax-oriented queries, without any representation of the meanings of these symbols in the machine doing the language processing itself.

By contrast, the Holy Grail of intelligent language processing would be the ability for the machine itself to process linguistic symbols with a semblance of understanding to do the processing regarding the meanings of the symbols being processed. This is what we can refer to as *semantics-driven language processing*, that we, humans do, when we use language with understanding. The proposed project will address this intuitive idea of intelligent language use

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 117-123, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110710

through tackling the conceptual and practical issues involved in understanding how natural language processing works for us, humans, the natural users of our natural languages as the medium of human thought, understanding and reasoning, and the applicability of the lessons learned from this study to artificial intelligence in computer science. Therefore, the two tasks indicated in the title of this project do require such broad-based collaboration: dealing with *natural logic* and *artificial intelligence* necessitates the recruitment of scholars from the wide range of fields indicated above.

Logic, as taught and practiced today as an academic discipline in the Frege-Russell tradition, is still primarily a formal mathematical study of certain fundamental forms of reasoning whose validity hinges on the fixed meanings of so-called "logical constants" or "logical connectives", such as those expressed by our languages' syncategorematic terms, like 'and', 'or', 'if … then', 'not', 'some', or 'every', which provide the formal structure of various forms of valid reasoning about any type of material expressed by our categorematic terms, namely, the terms that function as the subject or predicate of our categorical propositions.

However, classical formal logic (that is, standard predicate logic) is known to diverge from natural languages on various levels.

2.1. Mismatch of syntax

First, there is a known mismatch between the syntax of predicate logic and natural languages, based on predicate logic's treatment of all common categorematic terms as predicates (hence the name, "predicate logic") of singular referring expressions (individual names and variables, meant to represent proper nouns and pronouns of natural languages, respectively). Accordingly, predicate logic does not acknowledge the role of common terms in their referring function, forcing a reinterpretation of simple categorical propositions as conditionals or conjunctions, or leading to mere bewilderment over certain (so-called "pleonotetic") phrases. For example, on this approach, 'Some/All/Most Greeks are mortal' would turn into 'Some x is a Greek and x is mortal', and 'Every x is such that if x is Greek, then x is mortal', and just a source of embarrassment in the last case (and so also a motivation for generalized quantification theory), respectively. For more on this, see [1] and [3].

2.2. The divergence between formal and material validity

Besides this well-known mismatch in syntax, there is also the known fact that there are various valid forms of reasoning not captured by the notion of logical validity defined for the formal language of predicate logic, namely, those forms of reasoning whose validity is based on the information content of the categorematic terms of our propositions, which is precisely what is disregarded by the formal language. (For instance, "the page you are reading is in front of you; therefore, it is not behind you" or "it is white; therefore, it is not black" is a perfectly valid inference, the validity of which, however, is not captured by standard predicate logic.) Furthermore, there are obviously invalid forms of reasoning, which, however, based on their syntactical form alone, would appear to be instances of formally valid patterns of reasoning. (For example, "whatever is healthy is alive, but the food in this health food store is healthy; therefore, it is alive", which is invalid because of the equivocation of 'healthy' in the premises.)

This divergence between the notion of formal validity of a formal system and validity of actual pieces of natural language reasoning is nothing new. In fact, Aristotle ("the father of logic") recognized that his formal system of syllogistic reasoning did not capture all valid forms of reasoning (which is why he wrote the *Topics*, not surprisingly, in connection with the *Categories*, to deal with valid non-syllogistic reasoning), and that there were many invalid forms of reasoning

appearing to fit into valid syllogistic patterns (which is why he wrote his *Sophistical Refutations*, cataloguing various forms of fallacious reasoning).

It was also the realization of this divergence that motivated scholastic logicians' sophisticated discussions of the notion of logical consequence, striving to provide a unified account of valid reasoning in a "regimented" (explicitly regulated) version of academic Latin. But over the course of history's twists and turns, the scholastics' achievements were nearly completely forgotten, and were in modern times superseded by the theory and practice of formal logic as we know it. [8]

2.3. The need for a "natural logic"

Considerations of this sort recently more and more often prompted the expressed need for "a natural logic", both among philosophers of language and among computer scientists. ([6], [11]) A "natural logic" in the requisite sense would be a formal semantic system for a well-regimented fragment of a natural language with explicit phrase structure rules for its syntax and a recursively defined model for its semantics. (For an early attempt along these lines, see [5].)

What would make this approach "natural" in the first place would be cutting out the intermediary of a formal language along with its translation-rules (*à la* [10]) from the well-regimented fragment of natural language. In the second place, its semantics would allow the model-theoretical definition of a categorial structure, much in the vein of the Aristotelian theory of C*ategories*, licensing formal inferences based on the formal relations among the semantic contents of its categorematic terms, much in the vein of the *loci* of the Aristotelian *Topics*. Indeed, the compositional semantics for its propositions would enable the system to construct the *semantic content* of its propositions, thereby allowing for a content-sensitive definition of valid inference, based on the idea of the containment of the semantic content of the conclusion in the semantic content of the premises, along the lines of the conception of the *via antiqua* tradition of scholastic logic, yielding a relevant logic without the so-called paradoxes of entailment. [8] The recursive formulation of the formal semantics will allow the computability of the semantic values of any complex phrases for any arbitrarily chosen ontology; hence the system should easily offer itself for AI, enabling the computer to "see" the implications of all well-formed sets of sentences, which can actually be simply grammatical sentences of a natural language, thereby getting really close to "intelligent" natural language processing. (For the reason for the quotes, see, however [12]).

Approaching the natural logic in question from the starting point of standard predicate logic, the following steps need to be taken:

1. Represent a noun-phrase with a restricted variable, the values of which come out of the extension of the noun-phrase, provided it is not empty, otherwise its value is a zero-entity outside the universe of discourse. (Example: 'All humans are mortal' will become '(x.Hx)(Mx.)', where the values of 'x.Hx' will be elements of the extension of 'Hx', provided it is not empty, otherwise it is 0, an arbitrary item outside the universe of discourse.) The advantages of this approach include overcoming the mismatch of syntax mentioned above, the restoration of the full traditional Square of Opposition and syllogistic, and the immediate access to generalized quantification without the ontological extravaganzas of generalized quantification theory. A game-theoretical model for this approach as well as its perfect match with scholastic logic has been presented in [4].
2. Provide a tensed-modalized version of the previous system, in which the tense and modal operators in the matrices of restricted variables can perfectly model what scholastic logicians called the *ampliation* of terms: the phenomenon that in intensional contexts the

range of reference of our common terms becomes extended beyond actual existents, allowing quantification over and reference to non-existents. (Example: 'Every horse is alive; no dead horse is alive; therefore, no dead horse is a horse; however, whatever is a dead F was an F and was alive; therefore, a dead horse was a horse and was alive.) The advantages of this approach should be obvious to anyone who considers how ampliation in natural languages works.

3. Define a signification-function (inspired by Geach in [2]; see, however, [7] as well) as the semantic function of predicates in a model that contains at least actual and non-actual elements in the domain of discourse, thus: SGT(P)(u) is an element of the domain, and 'Px' is true (relative to f), just in case SGT(P)(f(x)) is an element of the actual part of the domain, where f(x) is the value of x in an evaluation f. [Example: 'Socrates is wise' will be true, just in case SGT('wise')(f('Socrates')) is an element of the actual part of the domain, i.e., just in case Socrates' wisdom is actual.] This is what medievalists refer to as the scholastics' "inherence theory of predication", the idea that the truth-maker of a simple predication is the actuality of the individualized property signified by the predicate in the individual(s) referred to by the subject. One immediate advantage of this approach is that it yields "fine grained intentions" even for logically equivalent predicates (SGT('triangular') does not have to be the same as 'SGT('trilateral') despite their equivalence), which then can be cashed out in intentional (psychological) contexts (for example: "John knows that a circle is a circle, yet he doesn't know that a circle is the locus of points equidistant from a given point", since he has no concept of a geometrical locus).
4. Choose a manageable "regimented" fragment of a natural language, and apply the semantic ideas listed above, including a categorization of its categorematic terms, allowing for "topical" inferences based on the significations the categorematic terms as well as on the syncategorematic structure of its propositions. To that end, define propositional significations compositionally, and allow mapping all semantic values of all your phrases onto a categorially structured ontology, as outlined in [9].

The rest is just a matter of deft programming, and you can have a machine that will "understand" your regimented linguistic fragment, insofar as it will have a semantic representation of a potential infinity of phrases generable in your fragment, and will, therefore, "intelligently" converse about the issues expressible in your fragment. The fragment can of course grow, and "cannibalize" ever larger portions of a natural langue; indeed, several natural languages, allowing for more intelligent translations than any syntax-driven systems can produce.

3. PRELIMINARY METHODOLOGY

To be able to test whether we are moving in the right direction, we need to start small. We should start out with a small vocabulary and a very restricted set of construction and interpretation rules, so we can see with our finite human intuitions that the rules we input into the machines do indeed produce the intuitively correct results. Still, we must do this with a view to the further ends. As should be clear from the foregoing, the natural logic to be taught to computers requires a well-regulated, regimented language, in which the terms themselves "bear their contents on their sleeves", namely, they carry essential information about the things they name. That is precisely the scenario, for instance, in organic chemistry, where the strictly regulated nomenclature serves exactly this purpose. So, taking some basic samples of texts in this field, we should first see if the results produced by the machine would fit our intuitive expectations, and whenever we find some anomalies, we need to chisel our rules accordingly. However, this "trial-and-error" period can be significantly shortened, and the whole enterprise can be substantially broadened by the input of our colleagues working in the fields listed above. So, the entire project needs to be given a more

definite shape through a launch event, where our collaborators can pitch in with their ideas from their fields of expertise.

4. THE RISKS AND THE BENEFITS

So, what if the production of genuinely intelligent, language using robots (as opposed to the glorified word processors of today) becomes a reality?

On the one hand, the exponential growth of the processing and storage capacities of today's computers and the similarly exponential growth of scientific data make it virtually impossible for our individual human intellects to keep pace with the explosion of scientific information. So, intelligent computers may soon surpass humans in many fields, especially those involving long, and boring tasks that humans would not and/or could not tackle, while robots would handle with ease and without complaints. So much for the benefits.

On the other hand, the potential for humans' being surpassed by robots in areas requiring intelligence immediately raises the spectre of the well-worn staple of Sci-Fi: the "killer bots". The usual Sci-Fi way to face these potential risks is to require the implementation of some "safety measures" (think Asimov's "laws of robotics"), and the plot usually unravels to show the ways in which those safety measures can be overridden by evil humans or can go awry through misinterpretation of their intent by merely "robotically intelligent" robots. The novel approach of this project would be based on the insight that intelligence and benevolence are not incompatible; indeed, on the contrary, ideally, they would go together. But if artificial intelligence is the artificial implementation of the ideals of human intelligence, then it should be the implementation not only of the theoretical, intellectual ideals (flawless calculation and reasoning), but also of the moral ideals (acting for the perfection of humanity in each individual and all of mankind). Thus, one of the tasks of this project will also be the articulation of these ideals, along with the means of their implementation.

5. LAUNCH EVENT

Accordingly, the launch event should be a workshop of invited participants who can provide their input concerning the requisite tasks and sub-tasks of the project. Who will oversee the fragment of a natural language (English) to be processed? Who will work on its semantics? What type of semantics should it be? What sort of ontology would it require? What sorts of categories will it include? What will be the basis of the categorization? What types of inference rules will the semantics license besides those licensed by syncategorematic structure? What are the "topical" rules of inference we can use from scholastic logic? How shall we treat fallacious forms of reasoning (that scientific texts are obviously not immune to)? What are the expectable risks and benefits even in the early stages of the project? In what ways will the resulting artificial "intelligence" be different from human intelligence, and what are the potential security and ethical issues emerging from them? This is just a somewhat random sampling of the theoretical and practical questions such a workshop will have to address.

6. TOPIC LIST FOR THE WORKSHOP

- natural logics, their semantics, proof systems and computational feasibility
- the development of natural logics: the Aristotelian and medieval logic tradition, and the later connections to the relational and quantifier logics of De Morgan, Frege, Russell and Peirce.
- natural logic as extended syllogistic logics

- natural reasoning and domain specific reasoning, e.g., handling of plurals and partonomies.
- natural logics and diagrammatic logics
- relationships between natural logics and natural languages (in the plural)
- semantic issues such as intensionality, and collective vs. distributive readings in natural logics
- natural logic for knowledge bases and AI systems, e.g., computational natural language processing
- formal ontology and category theory
- categories and topical reasoning in natural logic
- connections between natural logics and description logics
- security and ethical issues related to the emergence of "intelligent robots".

ACKNOWLEDGEMENTS

The idea of this workshop came up in collaboration with a Danish group of computer scientists working with Jørgen Fischer Nilsson,jfni@dtu.dk, who generously invited me for a discussion of the plans outlined above in 2018. For various practical reasons (including Jørgen's retirement and my getting married across two continents) the project eventually failed to materialize in the year we planned it for. This is just my attempt to revive what still appears to me a worthy idea.

REFERENCES

[1] G. Boolos: "To Be Is to Be a Value of a Variable (or To Be Some Value of Some Variable)", *The Journal of Philosophy*, 8(1984), pp.430-431.

[2] Geach, P. T. "Form and Existence," *Proceedings of the Aristotelian Society* 55(1954–1955); reprinted in his *God and the Soul* (London: Routledge & Kegan Paul, 1969) 42–64.

[3] Klima, G. (1990) "Approaching Natural Language via Medieval Logic", in: J. Bernard-J. Kelemen: *Zeichen, Denken, Praxis*, Institut für Sozio-Semiotische Studien: Vienna, pp. 249-267.

[4] Klima, G. and Sandu, G. (1990) with "Numerical Quantifiers in Game-Theoretical Semantics", *Theoria*, 56, pp. 173-192.

[5] Klima, G. (1991) "Latin as a Formal Language: Outlines of a Buridanian Semantics", *Cahiers de l'Institut du Moyen-Âge Grec et Latin*, Copenhagen, 61, pp. 78-106.

[6] Klima, G. (2010), "Natural Logic, Medieval Logic and Formal Semantics", *Magyar Filozófiai Szemle*, 54(2010), pp. 58-75.

[7] Klima, G. (2015) "Geach's Three Most Inspiring Errors Concerning Medieval Logic", *Philosophical Investigations*, 38(2015), pp. 34-51. Online "early view" DOI: 10.1111/phin.12075

[8] Klima, G. (2016) "Consequence", in Read, S.L.-Dutilh-Novaes, C., *The Cambridge Companion to Medieval Logic*, CUP: Cambridge, UK, pp. 316–341. doi:10.1017/CBO9781107449862.014

[9] Klima, G. (2021) "Form, Intention, Information: from Scholastic Logic to Artificial Intelligence", in Ludger Jansen & Petter Sandstad (eds.) *Neo-Aristotelian Perspectives on Formal Causation*, Routledge, 2020. ISBN: 9780367341206; https://www.routledge.com/Neo-Aristotelian-Perspectives-on-Formal-Causation/Jansen-Sandstad/p/book/978036734Bene1206

[10] Montague, R. (1974) "English as a Formal Language", in: R Montague: *Formal Philosophy*, Yale University Press, New Haven-London.

[11] Nilsson, J. F. (2015). "In Pursuit of Natural Logics for Ontology-Structured Knowledge Bases". In *Proceedings of the 7th International Conference on Advanced Cognitive Technologies and Applications* (COGNITIVE 2015) (pp.42-46). IARIA

[12] Stephan, K. and Klima, G. (2020) with "Artificial Intelligence and Its Natural Limits", AI & Society, 36(2020), pp. 1-10; Springer online first: DOI: https://doi.org/10.1007/s00146-020-00995-z; read online: https://rdcu.be/b4wmM

AUTHOR

Gyula Klima is a Professor of Philosophy at Fordham University in New York City and the Director of the Research Centre for the History of Ideas at the Institute for Hungarian Research in Budapest. He is specialized in ontology and logical semantics and works on comparative analyses of the relevant scholastic and contemporary theories**.**

DOUBLE MULTI-HEAD ATTENTION-BASED CAPSULE NETWORK FOR RELATION CLASSIFICATION

Hongjun Heng[1] and Renjie Li[2]

[1]Department of Computer Science and Technology,
Civil Aviation University of China, Tianjin, China
[2]Sino-European Institute of Aviation Engineering,
Civil Aviation University of China, Tianjin, China

ABSTRACT

Semantic relation classification is an important task in the field of nature language processing. The existing neural network relation classification models introduce attention mechanism to increase the importance of significant features, but part of these attention models only have one head which is not enough to capture more distinctive fine-grained features. Models based on RNN (Recurrent Neural Network) usually use single-layer structure and have limited feature extraction capability. Current RNN-based capsule networks have problem of improper handling of noise which increase complexity of network. Therefore, we propose a capsule network relation classification model based on double multi-head attention. In this model, we introduce an auxiliary BiGRU (Bidirectional Gated Recurrent Unit) to make up for the lack of feature extraction performance of single BiGRU, improve the bilinear attention through double multi-head mechanism to enable the model to obtain more information of sentence from different representation subspace and instantiate capsules with sentence-level features to alleviate noise impact. Experiments on the SemEval-2010 Task 8 benchmark dataset show that our model outperforms most of previous state-of-the-art neural network models and achieves the comparable performance with F1 score of 85.3% in capsule network.

KEYWORDS

Relation Classification, Double Multi-head Attention, Auxiliary BiGRU, Capsule Network.

1. INTRODUCTION

Relation classification is the one of important tasks of Nature Language Processing (NLP), its purpose is to recognize the semantic relation between marked entities in sentence [1], which is premised on entity recognition tasks. For example, in sentence "The suspect dumped the dead <e1>body</e1> into a local <e2>reservoir</e2>.", relation classification is to automatically identify the relation "Entity-Destination" expressed by the given entity pairs marked with HTML. In the field of application, relation classification can be used to enhance the existed knowledge base and create knowledge graphs or ontology knowledge base, from which users can retrieve and use the required knowledge. In addition, relation classification is also widely used in question answering system [2], textual entailment [3] and so on. Accurate relation classification can provide better quality for the above tasks.

Early relation classification methods mainly use machine learning and feature design which usually relies on NLP tools and simple hand-crafted features [4] such as entities' type, distance of

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 125-140, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110711

entities and dependency relation path. Recently, deep learning methods such as Convolutional Neural Network [5] (CNN), Recurrent Neural Network (RNN) [6] and other neural network architecture have been widely used for relation classification, these methods do not need to design feature manually and bring a certain performance improvement. Among them, RNN can capture local and global dependency information through gate mechanism. Representative RNN models include Long Short-Term Memory (LSTM) [7] and Gated Recurrent Unit (GRU) [8], which show satisfactory performance in processing sequential tasks, such as machine translation, speech recognition and relation classification especially. However, current RNN models for relation classification only use single layer to capture context features in sentence [9], which could be not enough. Because current NLP models prove that deeper neural network has stronger capability to represent semantic information and improve performance, such as transformer [10], residual network [11], etc. Therefore, it is necessary to explore the deep RNN network structure and improve performance of relation classification.

In order to alleviate the unrelated noise, attention mechanism is introduced to relation classification, which can help focus on important words associated with relation between entities. Frequently used attention includes word-level attention [12] and hierarchical attention [13], the latter is a combination of word-embedding level attention and feature level attention. Besides, multi-head mechanism is also introduced so as to capture distinctive fine-grained features from different representation subspaces, such as self-attention scaled dot product model [13]. However, the above attention models only have one head or single-level multi-head, there is still room to explore multi-level multi-head mechanism, which may help to further capture more distinctive features from sentence. Because sentence in relation classification is normally short, multi-level multi-head is more helpful to explore useful fine-grained information.

Capsule network [14, 15] is a new type of neural network proposed in terms of interpretability in recent years. Different from the previous classification methods, the capsule network combines features into a vector, which is called an instantiated capsule, and classifies by maximizing the length of capsule. Relation classification model based on capsule network has been explored, including CNN-based and LSTM-based capsule networks [16, 17]. The latter performs better than the former, but it has disadvantage. LSTM-based capsule network instantiates capsule through each hidden state of LSTM, but not all hidden states contribute to relation classification. Although some researchers have introduced attention, they do not perform weighted fusion of hidden states, which results in invalid noise fused into capsule and increases the computational complexity of dynamic routing process.

Motivated by above works, we propose the double multi-head attention-based capsule network model for relation classification. In this model, we design an auxiliary bidirectional GRU (BiGRU) architecture to deepen network in time dimension and boost the performance of single BiGRU. Besides, we propose a double multi-head mechanism and decrease the complexity brought by multi-head through max-pooling. Then the word-level features are weighted and merged into sentence-level features. Finally, we instantiate capsules through sentence-level features learned from different representation subspaces, and classify with help of dynamic routing algorithm. The contributions in this article would be summarized as follows:

(1) Firstly, we propose a feature extraction model with auxiliary BiGRU, which can make up for the lack of feature extraction performance of single BiGRU.
(2) Then, we propose a kind of double multi-head attention which enables the model to obtain more distinctive information of sentence from different subspaces.
(3) Our capsule instantiation strategy alleviates the noise fed to capsule network and reduces the complexity of network.

(4) Experimental results on SemEval-2010 Task 8 dataset show that our model achieves a state-of-the-art result with an F1-score of 85.3% in the field of capsule network.

2. RELATED WORK

As one of the methods of supervised learning, the deep learning model can automatically extract hidden features from the input sentence without manually constructing features, so it has received extensive concern from researchers. [18] proposed a Factor-based Compositional Model (FCM), which decomposes annotated sentences and extracts features from them. [19] proposed an enhanced dependency path structure to learn semantic representation. [6] constructed relative dependency features to capture the long-distance relation between entities by using Stanford dependency analysis tools, and used bidirectional LSTM to learn the hidden features and constructed lexical and sentence level features for semantic representation of sentence. [5] proposed to use the Shortest Dependency Path (SDP) to exclude the influence of irrelevant words or phrases, and introduced the negative sampling method into the CNN model to distinguish the directionality of the relation. [20] proposed SDP-LSTM model which uses LSTM to learn subtree feature of root node of SDP. [21] proposed a method of data enhancement using SDP, which uses the inversion of SDP between head and tail entities to add new data.

Since attention mechanism was applied to natural language tasks [22], attention-based models have been widely used in relation classification. [23] proposed context selective attention, using lexical level attention to selectively focus on words related to the target entity; [9] proposed a LSTM model based on attention that focuses on and integrates the word level features extracted by LSTM; [12] proposed a structured recurrent neural network model, which introduces attention into each layer of cascaded RNN network to pay attention to different lexical level features; [13] proposed an attention-based LSTM model, and introduced the multi-head self-attention mechanism proposed by Google Brain [10] in the word embedding layer to capture the meaning between words. At the same time, they added an entity-aware attention after LSTM layer to introduce information about entity as prior knowledge.

Capsule network is proposed to solve the representation limits of CNN and RNN network [14]. [15] replaced the scalar-feature of CNN with capsule and max-pooling with dynamic routing, they achieved the best performance in handwritten digit recognition task. [24] proposed matrix-capsule with EM (Expectation Maximization) routing algorithm, and achieved good performance in shape recognition task. For NLP tasks, [25] and [26] explored capsule networks for text classification. [27] proposed RNN-based capsule network in sentiment analysis. [16] first applied capsule network model to relation extraction, and achieved state-of-the-art performance on distant supervision relation extraction. [17] proposed an attention-based dynamic routing algorithm, which selectively focuses on different capsules for classification.

In this article, we will apply capsule network to relation classification, and explore multilayer RNN architecture, multi-head attention mechanism and instantiation of capsule.

3. MODEL

In this section, we introduce capsule network model based on double multi-head attention in detail. As shown in Figure 1, our model consists of four parts: (1) **Input Representation** layer maps each word in sentence to a fixed-dimensional vector and concatenates other features including relative position and part of speech. (2) **Feature Extraction** layer extracts low-level features from sentence through bidirectional Gated Recurrent Unit (BiGRU), and establishes the dependency relation between words; This layer also uses auxiliary BiGRU to make up for the

lack of single BiGRU. (3) **Double Multi-Head Attention** layer calculates the attention weights of the corresponding low-level features, and then selects the most significant features through max-pooling. Double multi-head mechanism is used to capture distinctive fine-grained information from different representation subspace. (4) **Capsule Network** layer divides the sentence-level features of attention layer into low-level capsules, and merges them into high-level capsules (classification capsules) through dynamic routing. Finally, length of capsules is calculated for classification.

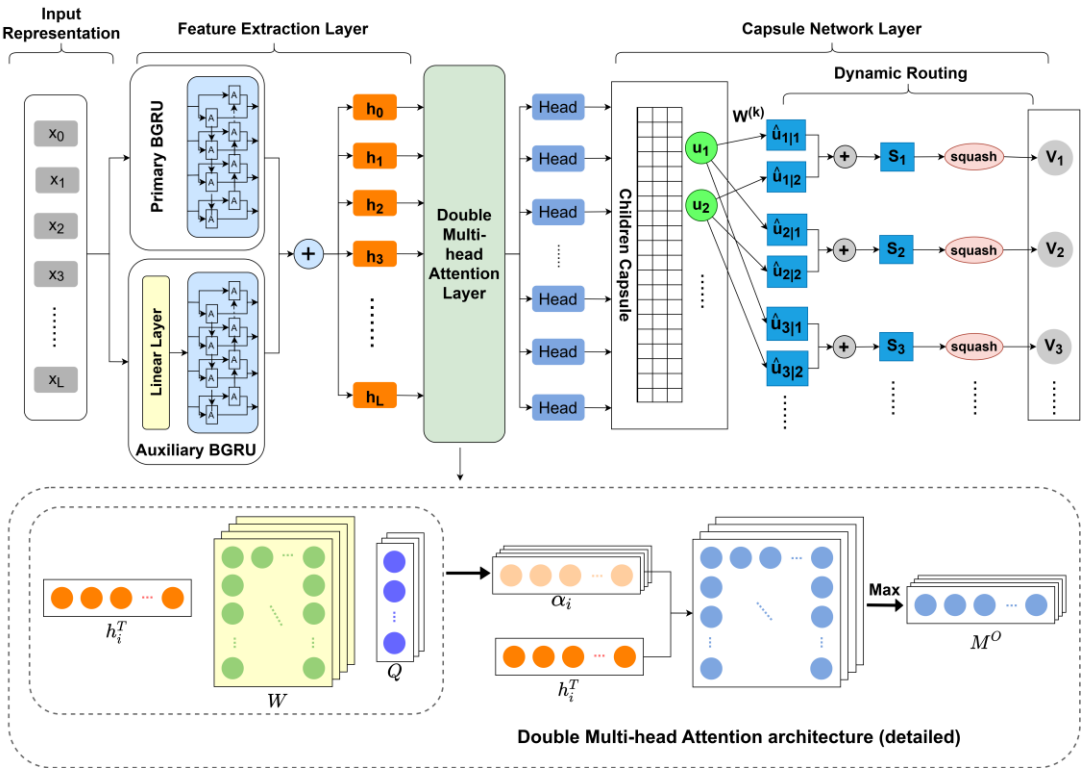


Figure 1. Double Multi-Head Attention-based Capsule Network Model

3.1. Input Representation

Word Embedding. Given a sentence $S = \{w_1, w_2, ..., w_n\}$ containing n words, we need to convert them into numbers that computer can recognize. Traditional method is encoding a word into a vocabulary-size vector through One-Hot, but this vector size is too large and there is no semantic correlation between words. Therefore, we adopt Word2Vec [28] proposed by Google. This method uses a word embedding matrix $W_{word} \in \mathbb{R}^{|V| \times d_w}$ to map each word to a low-dimensional dense vector that contains semantic meaning, where $|V|$ represents size of vocabulary and d_w is the dimension of word vector. In this article, the word embedding matrix is trained using the latest Wikipedia corpus, and the training model is Skip-gram. Finally, each word w_i in sentence is mapped to a vector $w_i^d \in \mathbb{R}^{d_w}$.

Position Embedding. In order to capture additional information about the relation between two target entities, we introduce position feature [29] to represent the relative distance between each word and two marked entities. For the given sentence in section 1, the relative distances between the word "dumped" and the two entities "body" and "reservoir" are respectively -4 and -7.

Therefore, the position embedding of each word w_i relative to two entities is expressed as $w_{i1}^p, w_{i2}^p \in \mathbb{R}^{d_p}$, where d_p is the dimension of the position embedding.

POS Embedding. Part of speech (POS) is the classification of word characteristics at the grammatical level. Adding POS features helps understand the attribute category of each word and identifies the relation between the components of sentence, and improves the robustness of the model. In our experiment, we use the NLTK tool to obtain the POS tags of words. The POS embedding of each word is represented as $w_i^{pos} \in \mathbb{R}^{d_{pos}}$, where d_{pos} is the dimension of the POS vector.

Finally, by concatenating these three types of features, the input representation of each word is $x = [w_i^d, w_i^{pos}, w_{i1}^p, w_{i2}^p]$, where position embedding and POS embedding are uniformly initialized by Xavier method [30].

3.2. Feature Extract

Recurrent neural network is a type of neural network with short-term memory capabilities, which has been widely used in natural language processing tasks. The simplest recurrent neural network only has one hidden layer, called a simple recurrent neural network [31]. However, it has long-term dependency problem and suffers from gradient vanishing and explosion [32], which causes the network to lose its ability to remember long-term information. To solve this problem, gated mechanism [7] is introduced gate to control the speed of information accumulation, including selectively adding new information and selectively forgetting previously accumulated information. The most representative gated recurrent neural networks are LSTM [7] and GRU [8]. Although both can solve the long-term dependency problem, GRU has one less gate than LSTM, and has a smaller computational complexity. Therefore, we use GRU for lexical feature extraction.

GRU controls the flow of information through reset gate and update gate. Note that the input of network is $x_t \in \mathbb{R}^{d_w+d_{pos}+2d_p}$, where t is current time step, $t \in \{1,2,...,L\}$ and L is length of sentence. $h_t \in \mathbb{R}^{d_h}$ is the hidden state at time t, where d_h is dimensionality of hidden state, h_t is updated by equation (1)-(4). Among them, r_t is reset gate that is used to control whether calculation of the candidate state $\tilde{h}_t$ depends on the state h_{t-1} at the previous moment; z_t is update gate that is used to control how much information the current state needs to retain from historical state, and how much new information needs to be received from the candidate state; W_i and U_i ($i \in \{r,z\}$) are weight matrices, $b_i (i \in \{r,z,h\})$ is bias, σ and tanh are sigmoid and hyperbolic function respectively; $\odot$ is element-wise product, which means the product of the corresponding elements of two matrices; The size of all state vectors is the same as h_t.

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (1)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (3)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (4)$$

For many sequential tasks, the current output is not only related to the past, but also related to the future. The bidirectional GRU enhances the capability of standard GRU by introducing a network layer that transmits information in reverse order of time. Therefore, we use BiGRU to capture the global sequential characteristics, final hidden state $\boldsymbol{h}_t^p \in \mathbb{R}^{d_h}$ can be expressed as $\boldsymbol{h}_t^p = [\vec{h}_t \oplus \overleftarrow{h}_t]$,

which is the element-wise sum of the forward state and the backward state. The size of the hidden state $\boldsymbol{h}_t^p$ is determined by hyperparameter d_h.

In recent years, deep neural networks have shown more excellent performance in tasks such as image recognition and machine translation. Inspired by the residual network [11], considering the characteristics of the strong fitting ability of RNN, we propose the parallel auxiliary recurrent neural network structure. Because RNN has a strong ability to fit data, the vertical stacking of traditional residual network structure is likely to cause serious overfitting, so the parallel structure is adopted to deepen the number of layers of the network at the time dimension. As shown in Figure 1, the feature extraction layer contains two layers of BiGRU. The upper layer is called primary BiGRU, which models the original sequence and outputs the hidden features $\boldsymbol{h}_t^p$. The lower layer is auxiliary BiGRU, which is merged into primary BiGRU in parallel to enhance the feature extraction performance of single BiGRU.

The auxiliary BiGRU layer consists of a linear layer, a BiGRU and a nonlinear activation function tanh. Auxiliary BiGRU receives the linear transformation of the input, after non-linear activation and encoding by itself, it outputs the hidden layer state $\boldsymbol{h}_t^a$, then accepts activation of *relu*, and finally is summed with $\boldsymbol{h}_t^p$. The above can be described as equation (5) and equation (6):

$$\boldsymbol{h}_t^a = BiGRU(\tanh(f(x_t))) \tag{5}$$

$$\boldsymbol{h}_t = \tanh(\boldsymbol{h}_t^p + relu(h_t^a)) \tag{6}$$

Where *BiGRU* means process of equation (1)-(4), f is linear transformation and $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$ is the output of feature extraction layer. The purpose of the auxiliary BiGRU is to learn the features lost by the primary BiGRU, and augments the capability of feature extraction layer.

3.3. Double Multi-head Attention

When bidirectional GRU deals with sequence, sentence can be encoded as a vector representation as time step t progresses. However, the length of the sentence is changeable. For a sentence that is too long, a single vector will lose the information at the head or tail of sentence. So, we retain the hidden vectors of BiGRU at each moment, and selectively weights and fuses the hidden features $H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_L\}$ through the attention mechanism. Attention distribution α_i of every hidden feature $\boldsymbol{h}_i$ can be expressed as equation (7):

$$\alpha_i = \text{softmax}(s(\boldsymbol{h}_i, q)) \tag{7}$$

Where q is relation query vector, $s(\boldsymbol{h}_i, q)$ is attention score function and softmax is used to normalize the score. We use bilinear attention score function, as shown in equation (8). Where W is a learnable bilinear matrix.

$$s(\boldsymbol{h}_i, q) = \boldsymbol{h}_i^T W q \tag{8}$$

In order to learn sentence from different subspaces, we introduce a double multi-head mechanism. Multi-head is introduced into bilinear matrix W and query vector q. The attention score function with double multi-head is shown in equation (9).

$$s(\boldsymbol{h}_i, Q) = \boldsymbol{h}_i^T W Q \tag{9}$$

Where $Q \in \mathbb{R}^{d_q \times d_c}$ is relation query matrix that is composed of d_c relation query vector with size of d_q; The matrix W becomes a three-dimension matrix from original two dimension, that means $W \in \mathbb{R}^{d_a \times d_h \times d_q}$, where d_a is a hyper parameter and represents number of multi-head.

Double multi-head mechanism is embodied by matrix W and Q, but it makes multi-head nested, which means the parameter amount of attention layer is increased to $d_a \times d_c$ times of original bilinear attention. In order to reduce complexity of network, we use maximum pooling operation, as shown in equation (10).

$$M^o = \max(\sum_{i=1}^{L} \alpha_i \boldsymbol{h}_i) \tag{10}$$

Where $M^o \in \mathbb{R}^{d_a \times d_h}$ is final output of double multi-head attention, the maximum pooling operation maximizes the weighted hidden layer features, which highlights the most salient features that have been paid attention. After fusion at each time t, d_a types of vector representations of sentence are output.

3.4. Capsule Network

Primary capsule: After the output of double multi-head attention, we need to resolve problem of how to instantiate capsule. For capsule network proposed by [15], capsule of a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity. In our work, capsule is the instantiation parameters of relation and built by sentence-level features. We combine d neurons into a capsule $u_i \in \mathbb{R}^d$, and obtain $d_a \times m$ primary capsule (children capsule) by splitting the matrix M^o where m is the division of d_h by d. The equation (11) is the list of children capsules:

$$U = [u_1, u_2, ..., u_{d_a \times m}] \in \mathbb{R}^{d_a \times m \times d} \tag{11}$$

$$\boldsymbol{u}_i = squash(u_i) = \frac{\| u_i \|^2}{0.5 + \| u_i \|^2} \frac{u_i}{\| u_i \|} \tag{12}$$

After obtaining the primary capsule, the length of capsule is squeezed into 0 and 1 by the activation of squash function in equation (12), because capsule network uses the length of capsule to represent the probability of relation classification.

Dynamic routing: The basic idea of dynamic routing is to map appropriate children capsules to parent capsules through non-linear loop iteration. We need a linear transformation on the children capsule to generate prediction vector $\hat{\boldsymbol{u}}_{j|i} \in \mathbb{R}^d$, where i and j are respectively children capsules and parent capsules. The linear transformation is realized by equation (13):

$$\hat{\boldsymbol{u}}_{j|i} = W_j^t \boldsymbol{u}_i + \hat{b}_{j|i} \tag{13}$$

Where $W_j^t \in \mathbb{R}^{I \times J \times d \times d}$ is a non-shared weight matrix and $\hat{b}_{j|i} \in \mathbb{R}^{I \times J \times d}$ is bias, I and J represent the number of children capsules and parent capsules respectively; Here $I = d_a \times m$ and J is the number of relation types.

See **Algorithm 1** and Figure 2 for dynamic routing, this algorithm controls the connection strength between children capsules and parent capsules through the coupling coefficient $c_{j|i}$ that is initialized uniformly, which means each children capsule is treated equally in first iteration;

then the coefficient is adjusted to select appropriate children capsules through later iteration. However, not all children capsules are effective for relation classification and there is still interference from noisy children capsules [26]. Therefore, we replace original softmax with leaky-softmax to update the connection strength, which is used to route noisy children capsules to additional dimensions.

Algorithm 1 Dynamic Routing Algorithm

1: **procedure** ROUTING $(\hat{u}_{j|i}, r, l)$
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{j|i} = 0$.
3: **for** r iterations do
4: for all capsule i in layer l: $c_{j|i} = \text{leaky-softmax}(b_{j|i})$
5: for all capsule j in layer $(l+1)$: $v_j = squash(\sum c_{j|i}\hat{u}_{j|i})$
6: for all capsule i in layer l and capsule j in layer $(l+1)$:
$b_{j|i} = b_{j|i} + \hat{u}_{j|i} \cdot v_j$
7: **return** $a_j = \|v_j\|$

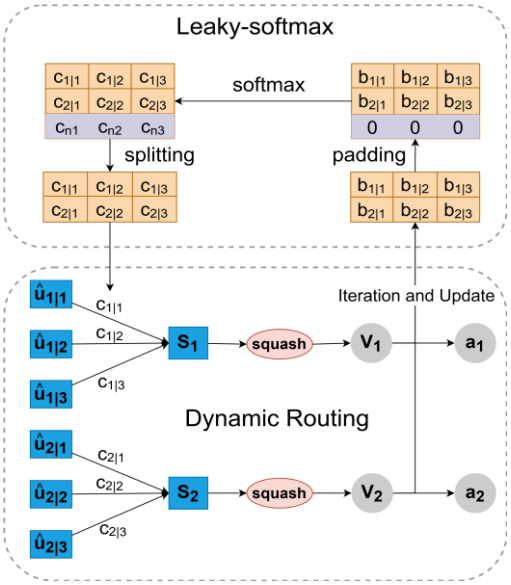


Figure 2. Dynamic routing with leaky-softmax

3.5. Training Procedure

For capsule network, the length of the instantiation vector v_j is used to represent the probability of relation. Unlike traditional neutral networks that use cross entropy loss, we use separate margin loss to calculate the loss L_j of each relation capsule *j*. In order to alleviate the overfitting of network, we use dropout [33] and L2 regularization. Dropout method improves the performance of the neural network by preventing the joint action of feature detectors during the forward propagation process. L2 regularization limits the weight update during the backward propagation process. In our model, we use dropout mechanism on the embedding layer and feature extraction layer. The loss function of each relation is shown in equation (14).

$$L_j = Y_j \max(0, m^+ - a_j)^2 + \lambda_1 (1 - Y_j) \max(0, a_j - m^-)^2 + \lambda_2 \| \theta \|_F^2 \quad (14)$$

Where $Y_j = 1$ if relation j is present; m^+ and m^- are threshold, λ_1 is the penalty rate for false positive and false negative, these three empirical parameters are usually set to 0.9, 0.1 and 0.5; λ_2 is coefficient of L2 regularization, θ represents weight parameters of our network (except for capsule network), $\|\cdot\|_F$ represents Frobenius norm. The total loss is sum of losses for all relations.

4. EXPERIMENT

4.1. Dataset and Experimental Setup

Dataset: Our experiment adopts the public dataset SemEval-2010 Task 8 [1], which contains 9 types of relation and an "Other" type. The 9 types are Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Entity-Origin, Instrument-Agency, Member-Collection, Message-Topic, Product-Producer; "Other" type is not of any of these nine types. In our experiment, we do not distinguish the direction of relations, so the total number of relations is 10. SemEval-2010 Task 8 dataset consists of 8000 sentences for training and 2717 sentences for testing. In order to compare our results with previous state-of-the-art models, we adopt precision P, recall R and F1 score to evaluate performance between our model and others. The definition of three metrics is shown in equation (15)-(17). The macro precision, macro recall and macro F1 are respectively the average of precision, recall and F1 of all relation categories.

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{17}$$

Setup: we randomly select 800 samples from the training set as development set for tuning hyperparameters. The best hyperparameters are shown in Table 1. Our experiment adopts 300-dimensional word vector pretrained in the latest Wikipedia corpus. Feature extraction layer of our model uses orthogonal initializer, other weights of network are initialized by Xavier method [30]. We train our model by Pytorch framework on platform Ubuntu and use one Geforce GTX 1650.

Table 1. Hyperparameters

Parameter	Description	Value
B	Size of batch	32
d_h	Hidden size of GRU	256
d_w	Size of word embedding	300
d_p	Size of position embedding	40
d_{pos}	Size of POS embedding	30
d_q	Size of query vector	17
d_a	Number of first head (W)	8
d_c	Number of second head (Q)	10
d	Size of capsule	8
lr	Learning rate	0.001
λ_2	Weight decay	0.0001
dropout	Embedding layer dropout	0.5
	Feature extraction layer dropout	0.1

4.2. Overall Experiment

Table 2 compares our double multi-head attention-based capsule network model with other four types of state-of-the-art models. Among the non-neural network models, the top is the support vector machine (SVM) [34]. This model uses manually created features and SVM classifier for relation classification, and achieves the best performance (82.2%) during the official competition. Models based on the Shortest Dependency Path (SDP) show excellent performance, including FCM [18], DepNN [19], depLCNN+NS [5], SDP-LSTM [20], BLSTM [6], DRNN [21]. SDP can ignore unrelated words between entities and construct a semantically directly related dependency path, which helps the model capture the dependency relationship between words more quickly. However, building of dependency tree often resorts to existing NLP tools, it is not always accurate and affected by sentence length, which costs time a lot. Introduction of attention has brought a very effective improvement to relation classification. By selectively assigning different weights, it highlights the most important words of sentence. Representative models are Hier-BLSTM [12], Att-BLSTM [9], Attention-CNN [23] and EA-BLSTM [13]. Recently, the capsule network model, which has received widespread concern in the field of image classification, has been used in NLP tasks, and a series of variants have been produced. Among them, [17] proposed a capsule network Att-CapNet, good results have been achieved with an F1 score of 84.5%.

Double multi-head attention-based capsule network proposed by us achieves an F1 score of 85.3% on SemEval-2010 Task 8 dataset. Although the performance of our model is not the best, it outperforms most other models without using external features like WordNet and SDP. Besides, compared with the capsule network relation classification model, our model achieves state-of-the-art result.

Table 2. Comparison with previous models on SemEval-2010 Task 8 (WAN represents words around nominals)

Models	**Macro F1(%)**
Non neural model	
SVM	**82.2**
SDP Model	
FCM	83.0
DepNN	83.6
depLCNN+NS	**85.6**
SDP-LSTM	83.7
BLSTM	84.3
DRNN	**86.1**
Attention-based Model	
Hier-BLSTM	84.3
Att-BLSTM	84.0
Attention-CNN	84.3
+WordNet, WAN	**85.9**
EA-BLSTM	84.7
Capsule Network Model	
Att-CapNet	84.5
Our model	**85.3**

For fair comparison with other models, we implement four of these models and use the same data pre-processing method and pretrained word vectors, which ensures that the input of each model is the same. Table 3 shows the result of precision, recall and F1 score of BLSTM [6], Att-BLSTM [9], Attention-CNN [23], Att-CapNet [17] and our model. It shows that the macro precision of our model is lower than that of Att-CapNet [17], but the macro recall exceeds others by 2.2%-4.1%, so the macro F1 score is increased by 0.9%-2.1%. According to the analysis above, we believe that our model is superior to the comparative models.

In order to explore the recognition effect of models on each relation, Table 4 lists the F1 score of five types of models for all relations (except Other). The comparison results show that our model is less effective in identifying "Component-Whole", "Entity-Origin" and "Member-Collection", F1 is lower than Att-CapNet and BLSTM. However, our model is better than other models in recognizing other relations, which has a greater contribution to the metric of macro-average F1 score.

Table 3. Fair comparison between our model and other four models

Models	**Macro P (%)**	**Macro R (%)**	**Macro F1 (%)**
BLSTM	81.7	87.3	84.3
Att-BLSTM	80.7	86.6	83.5
Attention-CNN	81.2	85.4	83.2
Att-CapNet	**82.4**	86.6	84.4
Our model	81.8	**89.5**	**85.3**

Table 4. Comparison of F1 (%) for each relation type

Relation Types	**BLSTM**	**Att-BLSTM**	**Attention-CNN**	**Att-CapNet**	**Our model**
Cause-Effect	92.9	90.7	91.4	92.0	**93.6**
Component-Whole	79.7	81.8	80.9	**83.3**	81.9
Content-Container	86.3	86.2	84.5	86.2	**86.8**
Entity-Destination	88.3	89.7	88.2	89.7	**91.0**
Entity-Origin	**85.8**	84.8	85.5	85.2	84.7
Instrument-Agency	74.7	72.8	73.5	74.2	**76.0**
Member-Collection	**85.1**	83.0	84.1	84.6	82.4
Message-Topic	83.0	84.5	82.7	85.5	**88.1**
Product-Producer	82.6	77.6	78.0	78.3	**83.4**

4.3. Ablation Study

In order to reflect the effects brought by auxiliary BiGRU, double multi-head attention and capsule instantiation strategy, we conduct an ablation study. The multiple variants derived from the model are shown in Table 5, we remove some components in our original model successively. "No multi-head (*W*)" and "No multi-head (*Q*)" respectively represent the situations of only removing multi-head of *W* and multi-head of *Q*; "No multi-head (*W* and *Q*)" represents removal of all multi-head of *W* and *Q* which means that our multi-head attention becomes the basic

bilinear attention. “No caps-ins-strategy” represents that we remove our capsule instantiation strategy.

Comparing the four models in the first, the second, the sixth and the last rows, it shows that our auxiliary BiGRU, double multi-head attention and capsule instantiation strategy effectively improve the overall performance. In specific, the auxiliary BiGRU boosts the precision P, double multi-head attention has a greater improvement in recall R, which slow down the impact of the decline in precision, so F1 is increased. Capsule instantiation strategy increases the precision. Comparing “No multi-head (*W*)”, “No multi-head (*Q*)” and “No multi-head (*W* and *Q*)”, results show that single multi-head does not improve the model, because improvement of recall is lower than impact of precision. But composition of these two multi-head brings an improvement of 0.6% for F1.

Table 5. Comparison with all variants in ablation study

Models	**Macro P (%)**	**Macro R (%)**	**Macro F1 (%)**
Our model (original)	81.8	89.5	85.3
No auxiliary BiGRU	80.8	89.6	84.9
No multi-head (*W*)	80.7	88.4	84.3
No multi-head (*Q*)	80.4	87.9	83.9
No multi-head (*W* and *Q*)	83.0	85.7	84.3
No attention layer	81.8	85.5	83.5
No caps-ins-strategy	78.9	85.9	82.2

4.4. Analysis of Double Multi-head Attention

Local analysis: Local analysis is to understand how the model makes decisions for a certain sample or group of samples. Figure 3 visualizes the attention weight α_i (see equation (7)) through the heat matrix diagram, which shows the importance of different words in a sentence for relation classification. The greater the importance of the word, the greater the attention score given to it, and the darker the corresponding colour in heat map. The 8 sub-graphs in Figure 3 respectively represent 8 heads on the bilinear matrix W, the vertical axis of sub-graph shows the 10 heads on the query matrix *Q*, the horizontal axis represents words in sentence, and the colour-bar on the right side indicates the size of attention score, which is between 0 and 1.

In order to explain the double multi-head attention more clearly, we only focus on the darker colour of each head (attention score greater than 0.6). Take the sentence provided in Figure 3 as an example, the two entities “survivors” and “houses” express relation “Entity-Destination”; In sub-figure (a), the 10 heads mainly focus on “into”; For other sub-figures, “moved”, “survivors” and “houses” are the main objects focused by attention layer. We can find that multi-head attention proposed in this paper focuses on the two entities and words expressing their relation, which is consistent with the focus of human. Therefore, our attention model finally distinguishes the sentence as relation “Entity-Destination”.

Global analysis: Global analysis is to explain the semantically meaningful components in the model and to understand how the model makes decisions on the entire dataset. According to the local analysis method, we extract the words (except for two entities) in sentence on the Testing set, and performs statistics according to the 9 types of relations; Due to space limitation, only the top four words with the largest frequency in each relation type are given, the statistical results are

shown in Table 6. We can find that for each type of relation, our attention model can identify the key words that express their relation. For example, the key words expressing the relation "Entity-Origin" are "derived", "from", etc., and for relation "Message-Topic", they are "about", "on", etc. In addition, we also count the proportion of entity pairs that our attention focuses on under each type of relation, and Table 6 shows that more than 90% of entity pairs can be captured by our attention. Thus, our double multi-head attention can identify the common patterns (feature words) of specific relation, and provides strong support for model's further decision-making.

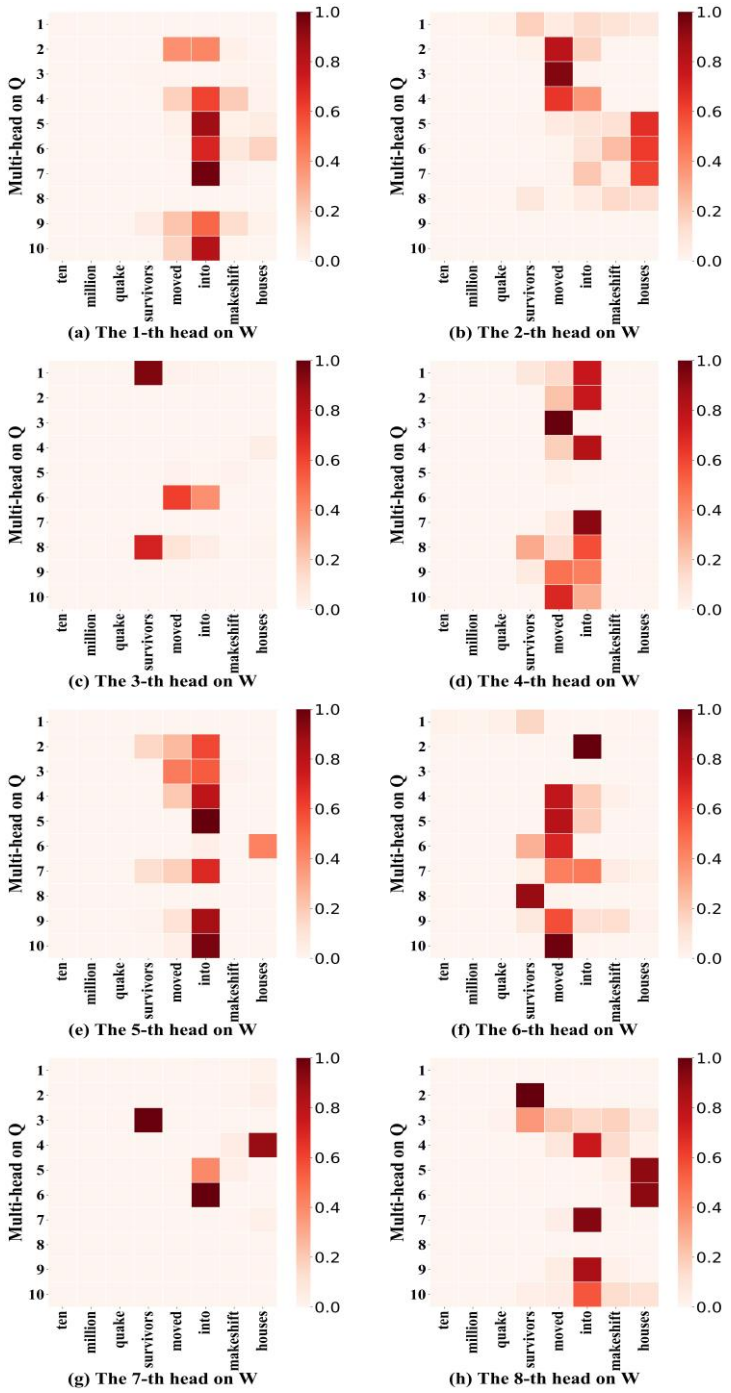


Figure 3. Heat Map of attention weight matrix for sentence "ten million quake Survivors moved into makeshift houses"

Table 6. The top four words with highest frequency and rate of entity pairs focused by the attention layer

Relation Types	Words focused	Rate of Entity pairs
Cause-Effect	Caused, by, from, cause	91.5%
Component-Whole	Of, with, in, has	96.2%
Content-Container	In, was, inside, with	80.2%
Entity-Destination	Into, to, put, in	88.4%
Entity-Origin	from, derived, of, away	95.0%
Instrument-Agency	With, using, of, by	98.1%
Member-Collection	Of, in, into, was	96.6%
Message-Topic	In, to, on, about	84.3%
Product-Producer	By, of, from, with	99.6%
Total	-	92.1%

5. CONCLUSIONS

We propose a double multi-head attention-based capsule network model for relation classification and auxiliary BiGRU that improves capability of single BiGRU for feature extraction. Our model achieves F1 score of 85.3% on SemEval-2010 Task 8 dataset using only word embedding, relative position embedding and POS embedding, and outperforms most of previous study. Ablation study shows that proposed auxiliary BiGRU, double multi-head attention and capsule instantiation strategy are effective. In addition, we analyse how the double multi-head attention highlights the words that contribute to relation classification from the local and global perspectives, as well as the common pattern recognition mechanism for specific relation types. In the future, we will use large-scale pre-trained language models such as Bert to further improve performance, and explore the potential of our model in the joint extraction of entity and relation as well as event extraction.

ACKNOWLEDGEMENTS

We would like to express our gratitude to the anonymous reviewers for their hard work and kind comments, which will further improve our work in the future.

REFERENCES

[1] I. Hendrickx et al., (2010) “SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals”, in Proceedings of the 5th International Workshop on Semantic Evaluation, pp33–38.

[2] D. Ravichandran and E. Hovy, (2002) “Learning surface text patterns for a question answering system”, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, No. July, pp41–47.

[3] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola, (2004) “Scaling web-based acquisition of entailment relations”, in Proceedings of EMNLP, Vol. 4, No. March, pp41–48.

[4] F. M. Suchanek, G. Ifrim, and G. Weikum, (2006) “Combining linguistic and statistical analysis to extract relations from web documents”, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Vol. 2006, pp712–717.

[5] K. Xu, Y. Feng, S. Huang, and D. Zhao, (2015) “Semantic relation classification via convolutional neural networks with simple negative sampling”, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp536–540.

[6] S. Zhang, D. Zheng, X. Hu, and M. Yang, (2015) “Bidirectional long short-term memory networks for relation classification”, in Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, pp73–78.

[7] S. Hochreiter and J. Schmidhuber, (1997) "Long short-term memory", Neural Comput., Vol. 9, No. 8, pp1735–1780.
[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, (2015) "Gated feedback recurrent neural networks", in Proceedings of the 32nd International Conference on Machine Learning, Vol. 3, pp2067–2075.
[9] P. Zhou et al., (2016) "Attention-based bidirectional long short-term memory networks for relation classification", in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp207–212.
[10] A. Vaswani et al., (2017) "Attention is all you need", in Proceedings of the 31st International Conference on Neural Information Processing Systems, pp6000–6010.
[11] Y. Y. Huang and W. Y. Wang, (2017) "Deep residual learning for weakly-supervised relation extraction", in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp1803–1807.
[12] M. Xiao and C. Liu, (2016) "Semantic relation classification via hierarchical recurrent neural network with attention", in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp1254–1263.
[13] J. Lee, S. Seo, and Y. S. Choi, (2019) "Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing", Symmetry (Basel)., Vol. 11, No. 6.
[14] G. E. Hinton, A. Krizhevsky, and S. D. Wang, (2011) "Transforming auto-encoders", in Proceedings of the ICANN, Vol. 6791, pp44–51.
[15] S. Sabour, N. Frosst, and G. E. Hinton, (2017) "Dynamic routing between capsules", in Proceedings of the International Conference on Neural Information Processing Systems, pp3859–3869.
[16] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang, and H. Chen, (2018) "Attention-based capsule networks with dynamic routing for relation extraction", in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp986–992.
[17] X. Zhang, P. Li, W. Jia, and H. Zhao, (2018) "Multi-labeled relation extraction with attentive capsule network", in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp7484–7491.
[18] M. R. Gormley and M. Dredze, (2014) "Factor-based compositional embedding models", in NIPS Workshop on Learning Semantics.
[19] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, and H. Wang, (2015) "A dependency-based neural network for relation classification", in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 2, pp285–290.
[20] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, (2015) "Classifying relations via long short term memory networks along shortest dependency paths", in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp1785–1794.
[21] Y. Xu et al., (2016) "Improved relation classification by deep recurrent neural networks with data augmentation", in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp1461–1470.
[22] D. Bahdanau, K. Cho, and Y. Bengio, (2015) "Neural machine translation by jointly learning to align and translate", in Proceedings of the 3rd International Conference on Learning Representations.
[23] Y. Shen and X. Huang, (2016) "Attention-based convolutional neural network for semantic relation extraction", in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp2526–2536.
[24] G. Hinton, S. Sabour, and N. Frosst, (2018) "Matrix capsules with EM routing", in Proceedings of the 6th International Conference on Learning Representations.
[25] L. Xiao, H. Zhang, W. Chen, Y. Wang, and Y. Jin, (2018) "MCApsNet: Capsule network for text with multi-task learning", in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp4565–4574.
[26] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, (2018) "Investigating capsule networks with dynamic routing for text classification", in Proceedings of the 2018 Conference on Empirical Methods in Natural Language, pp3110–3119.
[27] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, (2018) "Sentiment analysis by capsules", in Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp1165–1174.
[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, (2013) "Efficient estimation of word representations in vector space", in Proceedings of the 1st International Conference on Learning Representations.

[29] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, (2014) "Relation classification via convolutional deep neural network", in Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, pp2335–2344.
[30] X. Glorot and Y. Bengio, (2010) "Understanding the difficulty of training deep feedforward neural networks", in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Vol. 9, pp249–256.
[31] J. Elman, (1990) "Finding structure in time", Cogn. Sci., Vol. 14, No. 2, pp179–211.
[32] Y. Bengio, P. Simard, and P. Frasconi, (1994) "Learning long-term dependencies with gradient descent is difficult", IEEE Trans. Neural Networks, Vol. 5, No. 2, pp157–166.
[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, (2013) "Improving neural networks by preventing co-adaptation of feature detectors", Computer Science, Vol. 3, No. 4, pp212-223.
[34] B. Rink and S. Harabagiu, (2010) "UTD: Classifying semantic relations by combining lexical and semantic resources", in Proceedings of the 5th International Workshop on Semantic Evaluation, pp256–259.

AUTHORS

Hongjun Heng received the Ph.D. degree from Nankai University under the supervision of Prof. Z. Wang. He is currently an Associate Professor with the Department of Computer Science and Technology, Civil Aviation University of China. His research interests include intelligent information processing and computer application, specifically include natural language processing and knowledge graph.

Renjie Li received the B.S. degree from the Sino-European Institute of Aviation Engineering, Civil Aviation University of China, China, in 2018, where he is currently pursuing the M.S. degree. His current interests include knowledge graph and nature language processing.

BASKETBALL-51: A VIDEO DATASET FOR ACTIVITY RECOGNITION IN THE BASKETBALL GAME

Sarbagya Ratna Shakya,, Chaoyang Zhang and Zhaoxian Zhou

School of Computing Sciences and Computer Engineering,
University of Southern Mississippi, Hattiesburg, USA

ABSTRACT

In recent years, there has been an increase in the association of technology in sports and live sports broadcasting networks. From score updates, broadcasting commercials, assisting referees for decision making, and minimizing errors, the adoption of technology has been used for fair play and improve results. This has been possible with the advancement in video analysis, classification techniques, and the availability of resources. This paper introduces a new labelled video dataset collected from a live basketball game broadcasted on live TV to determine the type of basket scored in the basketball game. Among different shots, the points the player can score are basically of three types: 3 points, 2 points, which depends on the range of shots taken and 1 point which is the free shots taken after a foul. This dataset consists of labelled video clips collected from the live broadcast of the game from the broadcasting medium to classify different scoring activities. This paper also gives the preliminary analysis of the dataset for different class labels using 3D ConvNet and two-stream 3D ConvNet methods to show the complexity of the dataset.

KEYWORDS

Basketball dataset, 3D ConvNet, two-stream 3D ConvNet.

1. INTRODUCTION

With recent development in the field of Artificial Intelligence (AI), computer vision, and innovations in deep learning and neural network algorithms, the application of these techniques in different fields especially in sports has been increasing at an incredible pace. Areas of sports such as monitoring player fitness, player injury detection, sports marketing, broadcasting, wearable technology, have been using AI and computer vision in the past few years. Sports personnel, TV broadcasters have implemented AI in automated journalism to enhance sports coverage and spectator experience. Also, AI-powered Wearable devices worn by players provide data that can be used in player tracking [1], player performance analyzing, and optimizing training and player efficiency. With the use of sensors and tracking devices, only limited data about the player and game can be obtained and will be hard to analyze these data in real-time. Also, the necessity to place sensors on the body of players while playing has made it unrealistic to collect data in a real-world scenario. These drawbacks with the sensor data can be resolved if videos can be used to extract information rather than using sensors. But with the large video data collected for many hours and numerous replays to extract the required information from it can make it tedious for people in analyzing the footage. With the application of computer vision, an automated system can be developed that can analyze the videos, players, game situation and gather important insights and valuable analysis from them. Ball tracking, player tracking is some of the major application of computer vision implemented that has provided coaches better understandings of the formation of teams and given instant analysis to better the performance.

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 141-153, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110712

The application of AI-based techniques has already been implemented in different sports[2]. AIautomated video highlights generation and broadcasting which picked key moments of the game has been announced and on development. In sports like tennis the ball tracking system has been used to identify whether the ball has landed in or out of bounds. It uses computer vision to construct the trajectory of the detected ball using multiple frames from multiple camera angles. Also, in soccer, the goal-line technology has been adopted that uses multiple camera systems where it uses the computer vision technology system to determine whether the ball has crossed the line on the goal line or not. The analysis gives the referee enough information to make decision in a quick time. This implementation of technology has helped the officials to take the quick and right decision and minimize human errors in sports. Although the application in different sports has provided encouraging results, it has been a challenge due to different factors, to fully automate these systems by using the video.

In this paper, we introduce a new video-based basketball dataset derived from live video broadcast TV for classifying scoring activity in the basketball game. The main objective of building this dataset is to develop a well-labeled dataset dedicated to the activities related to basketball as there is a lack of such datasets for applying automated computer vision techniques for activity recognition. Only a few in other sports like Volleyball[3] where specific sports-related action classes have been used for classification. Our dataset may help researchers to develop and test different architecture and algorithms to evaluate real-time basket and score recognition from live video of basketball games without any human interventions.

Most of the datasets currently existing for basketball have been the dataset that shows the statistics of players and coaches. Most of them include statistics like the number of games played, number of games won, offensive-defensive efficiency, field goal percentage, and many more. Although with the successful application of machine learning and deep learning in the field like face recognition, video segmentation, and its increased application in the video analysis field, the lack of large visual based basketball dataset has been a problem to apply classification techniques in this sport. This dataset provides a short 6 second length clips labelled with different scoring activities as the classification classes sourced from the live videos broadcast on TV. The dataset contains the clips of the scoring attempts from the players throughout the games divided into 8 different categories. Each clip contains information about whether it is attempted from long-range, mid-range, or short-range shots and if it has made or miss the basket. Hence it will help to gain information about the scoring points scored by the player whether it is three-point shots, two-point shots or a free throw for one-points. This should be helpful to develop an automated scoring system during a live game without any human involvement. Also, this will provide information to assist the referee during an unclear condition for better decision making. Hence, the dataset can provide necessary information for research to develop a fully automated system using only the commercial broadcasted video data without using any special camera or camera setup during the games. But the dynamic background of the clips with audience movement, the movement of the camera, change of camera angle, multiple movements of players, presence of flashing information/advertisement, and difficulty in tracking the ball in the background have been some of the challenges this dataset brings in classifying the videos.

Therefore, the main contributions of this paper can be summed up as follows:

a. A new labeled basketball-related action video dataset for activities related to scoring in the basketball games derived from the broadcasted video for a real-world scenario.
b. A baseline reference model and analysis with state of art action recognition models is done to set benchmarks on this dataset.

The organization of the rest of the paper is as follows. Section 2 shows some previous work, summarize existing benchmark sports dataset, as well as a literature review of some of the other sports-related dataset. Section 3 gives a brief explanation and details of the dataset, the data collection process, dataset features, and annotation framework. Section 4 presents the explanation of the baseline models being used, Section 5 shows some experimental results obtained for evaluating the performance of the deep learning models, while finally Section 6 contains the conclusions and future work lines.

2. LITERATURE REVIEW

Many video-based benchmarks human activity recognition dataset such as HMDB51[4], UCF101[5], ACTIVITYNET[6], KINETICS[7] has been published which contains sports-related actions such as catching or throwing a baseball, juggling a soccer ball, playing cricket, shooting a basketball, etc. and has been included as human action classes. These datasets contain videos and images of different activities performed by the subject inside the video and have contributed greatly towards the video classification, with the study of human action classification from clips of humans performing different sports-related activities.

Not only activity related but also some sports-specific video dataset has been collected for video analysis to extract some critical and beneficial information from the video content. Some benchmark datasets Wang et al.[8], UCF Sports[9], Olympic Sports[10], Sports -1M [11], SVW[12] for sports video analysis collected from different sources such as YouTube, TV broadcast, smartphone, and tablets have been developed. These datasets contain images and videos of different sports such as baseball, basketball, tennis, badminton, football, horse riding, running, etc., and have been used to classify different sports. But the lack of inclusion of all the actions related to the specific sports has made it difficult to apply in real-world application and classify the actions related to a variety of actions performed on a single sport. As per example, in dataset UCF101, there are only the activities related to the dunking action as Basketball Dunk and basketball shooting. Rather in the real game, many actions like dribbling, passing, and other different scoring actions are present which is lacking in the previously published dataset. The most published dataset consists of several videos range from around 100 to millions of videos of variable length. The class categories also range from a few to around 500 different sports classes focused on sports actions.

Some specific basketball-related datasets such as SportUV which is explained in this link (https://www.nbastuffer.com/analytics101/sportvu-data/) has been developed where the automated ID and tracking technology system records the tracking of the spatial position of the ball, players, and referee on the court 25 times per second during the game. This dataset indicates when the three-point shot is taken and whether the shot is successful. This data was joined with the play-by-play data from the NBA and is kept that are in both datasets. Since the 2017-2018 seasons, the Second spectrum has been the official player tracking technology provider that collects 3D spatial data of movements of ball, players, referee locations from cameras installed on NBA arenas. In APIDIS dataset [13] the video is collected from 2MPixels color cameras installed around the basketball court (four on each side). In [14] the authors present methods to predict the behavior of the basketball player from the first-person videos (10.5 hours) collected by the University team at Northwestern Polytechnical University.

In earlier times research has also been done to detect scores from the broadcast video in real-time. In [15], the author has proposed a real-time approach to detect score region and recognize the score in broadcast basketball video using frame difference and texture information. They have shown that their approach achieves high accuracy compared to traditional text recognition methods.

Many machines learning and deep learning architecture has been used for action recognition using these video datasets. Most video-based architecture used 2D and 3D content with LSTM[16][17][18][19][20][21] for transferring information across frames and capture long-range dependencies. In recent years, two-stream networks[22][23][24][25] which uses two different types of stream, RGB and flow data, are fused for prediction. For our analysis, we used the 3D ConvNet to train our model from scratch for both our RGB and optical flow video data along with two-stream 3D ConvNet with early fusion techniques. The detail of the dataset is explained in section 3 and the methods we used are explained in section 4.

3. DATA COLLECTION AND DATA CHARACTERISTICS

In this section, we describe how the data was collected, processed, and prepared.

3.1. Data Collection

Step1: Collection of live video games

The videos are collected from the live video of NBA basketball games broadcasted from the different broadcasting channels. Apart from players actions, the videos include scores, flashing information, replays, highlights, interviews, advertisements, and other graphics displayed during the game as it is being broadcasted on live TV. The quality of the video recorded at first is in HD.

Step 2: Manually store the label and timestamp.

Once the live video of the game is recorded, we then manually list the timestamp of the video at the point when the ball reaches near the rim after the player makes a shot towards the basket. The list consists of the hour (HH), minutes (MM), and second (SS) of the video at the time the ball is near the rim from the starting of the video. Then it is manually annotated whether the shot has made the basket or miss the basket and the range from which the shot was taken for labeling the clips.

Step3: Generate clips

Once we have the timestamp from the video, a 6-sec video clip was generated where the point of action is in the middle of the clips. This time duration is chosen so that the clips have complete information of the action such as from where the shot was taken, make or miss of the shot, and the information if there are any rebounds or multiple attempts to make the basket after the first shot. We generate the clips from videos of 51 full basketball games. The average number of clips generated from each video is about 200. To make it more appropriate for experimental purposes, the dimension of the clips is also reduced from HD to the dimension of Quarter Video Graphics display (QVGA) (320×240) pixels value. Sample video frames of the collected dataset are given in Figure 1. The figure shows the sample frame of videos classified into 8 different labels and three specific timestamps of the activities like the point of the throw, the point of the ball on the rim and the point after the ball make or miss the basket.

Step 4: Optical flow dataset

From the RGB video clips, for experimental analysis using temporal data, we also generate the optical flow videos for optical flow data. We find the relationships between the consecutive frames using the optical flow concept which was first proposed by[26] and generate the optical

Figure 1: Example of frames from the video. From left to right represents frames of videos at the time of the shot, ball on the rim, and the ball after the shot and from top to bottom represents the action for 2p0,2p1,3p0,3p1, ft0,ft1, mp0, and mp1.

flow video clips from the RGB video clips. For that, we used the open-source library OpenCV with the Gunnar Farneback optical flow technique [27]. This method detects the pixel intensity changes between the two consecutive frames and gives the highlighted pixels. The optical flow clips were generated and labelled from the original RGB clips. The objective of generating the optical flow clips is to learn the temporal flow of the information in the video.

3.2. Dataset Characteristics

This section describes the detailed characteristics and features of the dataset. The Dataset has a total of 10,311 video clips generated from 51 NBA basketball games broadcasted in the media. The videos are entirely from the third person view captured from the camera used from sports broadcasting media. The clips are initially labelled into 8 class labels: defined as two-point miss(2p0), two-point make(2p1), three-point miss(3p0), three-point make(3p1), free throw miss(ft0), free throw make(ft1), mid-range shot miss(mp0), and mid-range shot make(mp1). The make and miss of the shot taken by players has been represented by 1 and 0 in the activity labels. The distribution of the number of class labels is represented in figure 2. The highest number is that of the three-point miss (about 20% of total data) and the lower is for the mid-range make and miss (around 5.4% of total data). The highest different between the make and miss is in free-throw i.e., there is more free-throws make in games than free throw miss as shown in figure 2. Also, the difference between make and miss is lowest in the mid-range shot which has somewhat equal number of make and miss. To study the characteristics of the dataset, the analysis has been studies based on different groups. The dataset has been grouped on 4-class based on the range of shots taken. The four class categories are two-point, three-point, mid-point, and free throw. Again, to analyze the miss/make of the shots the whole dataset is divided into 2-label dataset of make and miss. The details about the experimental analysis of these groups are described in the experimental setup section of the paper. For proper grouping, the nomenclature of the clips is given to provide information about the label, video number, and the timestamp of the clip which represents the 3rd second of the clip. The optical flow clips were names similar to its RGD video clips and have the same characteristics. Identical models and parameters were used on the optical flow dataset to analyze the performance as has been used for the RGB dataset.

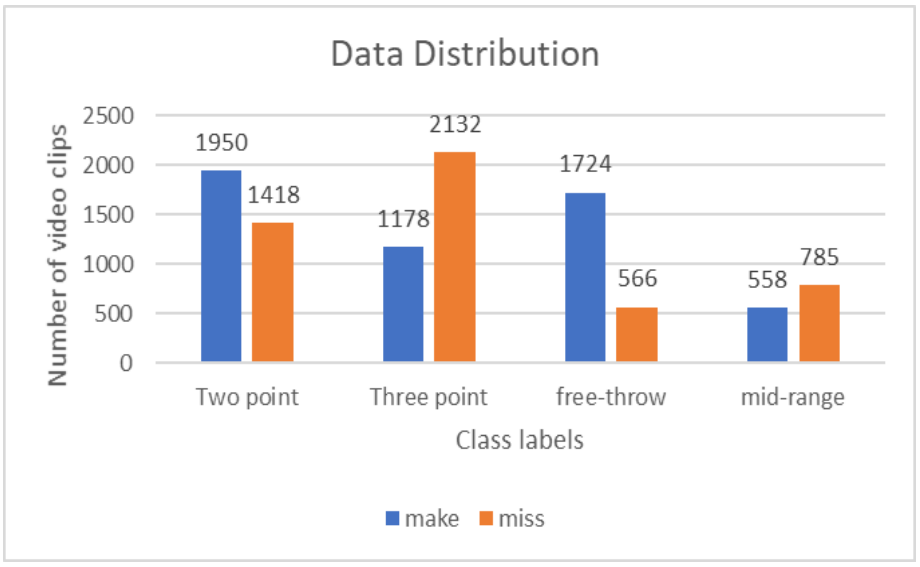


Figure 2: Example of data distribution of the dataset based on different groups.

4. BASELINE PERFORMANCE MODEL

For our evaluation, we used two state-of-art deep learning models for video classification: i) 3D convolution network [28] for video segmentation on both RGB and optical flow video datasets and ii) two-stream ConvNets[22]. The details of the approach are described as follows. In this section, we describe the basic overview of this architecture and explain how we apply this architecture in our experiments.

4.1. 3D ConvNet

3D ConvNets are like the 2D ConvNets but with three-dimensional convolutional kernels which can make segmentation prediction for a volumetric patch. Because of its 3D nature, it seems to be the ideal approach to video modeling and has been used for many video segmentation[29][30] approaches. In addition to the height and weight of the 2D convolutional kernel, the 3D ConvNet has third kernels representing the spatial-temporal filters. Hence it will help to analyze the spatial and spectral features of action between frames of the clips in time dimensions.

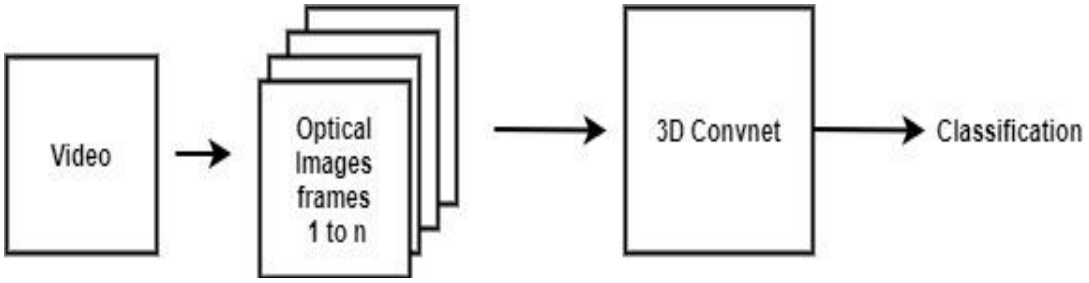


Figure 3: Example of the 3D ConvNet

For our analysis, we implemented a 3D convolutional neural network similar to C3D [31], which has 4 convolutional layers block which includes 3D CNN layer followed by max-pooling layers, dropout layers, and batch normalization layers. Then it is followed by 1 global average pooling layer and 3 dense layers. The model has the input of video clips with 50 frames with a pixel size reduced to size 80 × 80. The size of the frame is chosen to transform the high-dimensional data with minimum size videos to reduce the memory requirement and computational complexity of the model. The filter numbers range from 16, 64, 256, 512 for 3D CNN and 256 and 32 for fully connected layers. The SoftMax function is used to predict the output classes. The loss function used is categorical cross-entropy. For all our experiments we use the initial learning rate of 0.1 with Nadam optimizer with its default arguments parameters, batch size of 20, and train for 100 epochs. All models are trained on an Nvidia Titan Xp GPU.

4.2. Two stream 3D network

The main idea behind two-stream networks is to train two-stream of CNN networks, one RGB

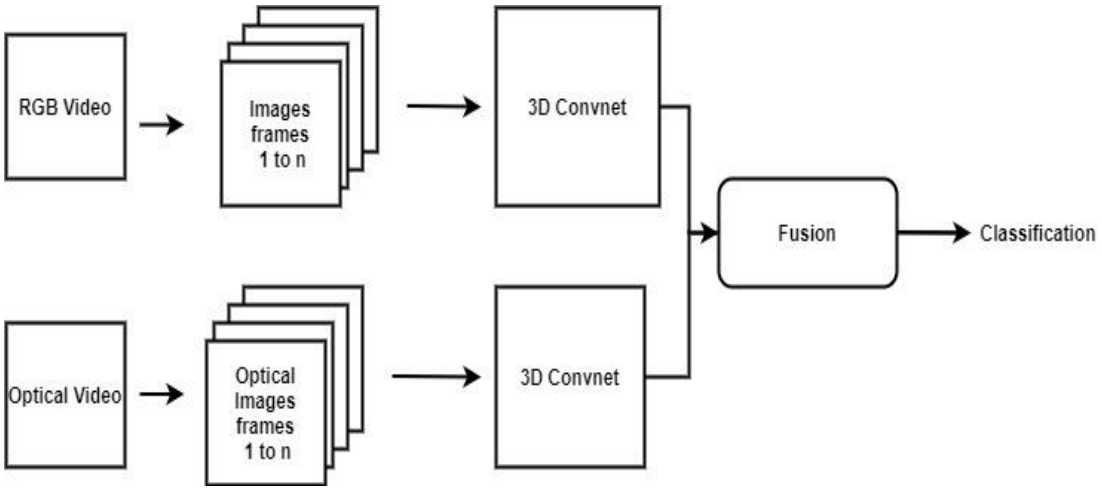


Figure 4: Example of two stream 3D ConvNet

data to get spatial information and another stream with optical flow data for temporal information. These two networks are then fused in some part of the model before classification. In Earlier approaches, a two-stream network is implemented by using short temporal snapshots of video by averaging the predictions from a single RGB frame and a stack of 10 extremely computed optical flow frames [22]. In our case, we generated the optical flow clips from our RGB data clips and used the optical flow clips as our input video data. For our experiment, we used the identical 3D ConvNet network with same input parameters for both the stream with RGB video as input data on one stream and the corresponding optical flow video on the second stream. We then fuse the output taking average after the 3D ConvNet layer and then pass it to the fully connected layers. The configuration of the two-stream network is as shown in figure 4.

5. EXPERIMENTS AND ANALYSIS

This section presents some evaluations using this basketball dataset to illustrate the characteristics and challenges for action recognition. We used one of the most used deep learning architectures, 3D ConvNet for video classification. The task aims to correctly classify the scoring label of the video clips that contain the scoring activity of the player when they take a shot. Here we use the clips derived from the broadcasted videos to train the classifier and evaluate the performance based on different classifier algorithms. For our evaluation, we test our model into two classification types: Subject dependent and subject independent. Here the subject represents the video of the basketball game. The objective of analyzing in different classification type is to study the impact in the performance of the model during training and testing because of the difference in the background audience, court color, players jersey, player movement and camera orientations that can have different features in different videos from one game to another.

5.1. Subject Dependent

In Subject Dependent classification, the division of the dataset is done randomly to training and testing dataset where both contains samples from all the subjects. If there are samples for n number of subjects, both training and testing dataset contains certain percentage of samples from all n number of subjects. For the subject dependent, the total

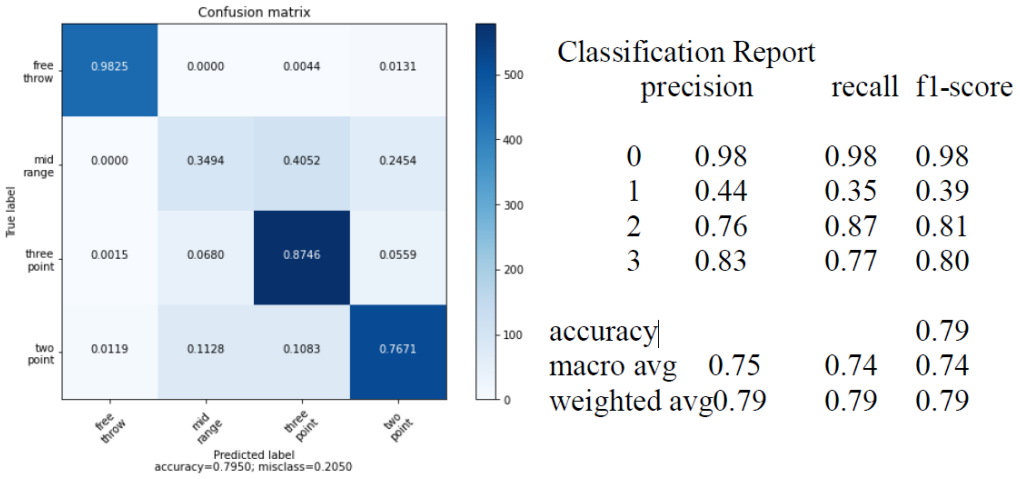
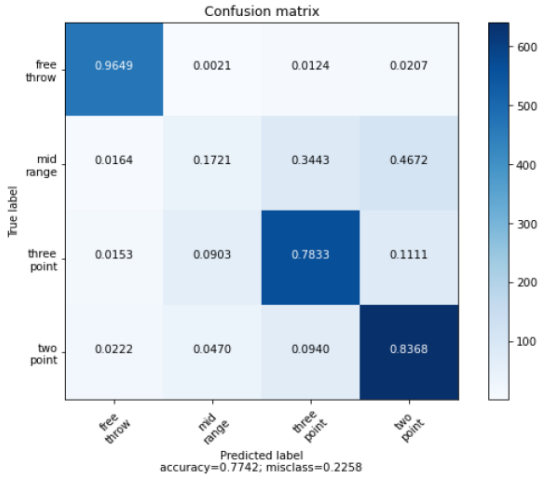


Figure 5: Confusion Matrix and classification report for the RGB basketball dataset using 3D ConvNet for 4 class subject dependent classification.

clips collected from 51 games is divided into (80/20) training and testing datasets based on their class labels where clips from each video can be in the training and testing data. This random division of the total data into training and testing data, have samples corresponding to the same video and, most likely the samples of all subjects. The confusion matrix and classification report for the RGB dataset for 4-class subject dependent classification is shown in figure 5. The 0,1 2 and 3 in classification reports represents the classification class of free throw, mid-range, three-points and two-points. Here, the mid-range has the lowest recall value of about 35% whereas the free-throw activity has the highest classification recall value of about 98%. The overall accuracy is 79%. Most of the mid-range shots has been misclassified as three-point and two-point shot as there is a small margin of range between mid-range with two-point and three-point range. Also, the low number of training data for mid-range can be the reason for low classification accuracy.

5.2. Subject Independent

In subject independent classification, the division of dataset is done to training and testing dataset such that samples from the subject included in training are not included in testing data. That isfor n number of subjects, samples from n-k subjects are used in training data whereas samples from the remaining k subjects are used in testing. The system classifier will be tested with testing data having completely new features than the training data. For subject independent classification, the clips from 41 different games were used as training while the clips from the next 10 games were used as testing datasets. This is done to make the subject and features of the testing data completely unknown to the classifier trained on entirely different training video data. The objective is to analyze the effect on the classification performance due to differences in a court appearance, player movement, background change, camera orientation,



Classification Report

	precision	recall	f1-score
0	0.94	0.96	0.95
1	0.29	0.17	0.22
2	0.78	0.78	0.78
3	0.76	0.84	0.80
accuracy			0.77
macro avg	0.69	0.69	0.69
weighted avg	0.75	0.77	0.76

Figure 6: Confusion Matrix and classification report for the RGB basketball dataset using 3D ConvNet for 4 class subject independent classification.

and other factors in testing data as compared to the dataset used for training. The confusion matrix and classification report for the RGB basketball dataset using 3D ConvNet for 4 class classification is shown in figure 6. Here also the mid-range has been misclassified mostly as two-point range and three-point range and has the lowest recall value of 17%. The overall accuracy is 77% which is less than the subject-dependent classification accuracy.

In both cases, 20% of the training dataset was used as a validation dataset for evaluating the performance of the model during training. With each epoch, the model tests the performance on the validation dataset and will save the model if the performance betters (validation loss improves) than the previous saved best model. We evaluate our training for 100 epochs. Thus, at the end of the 100 epochs, the saved model will be the best one with the lowest validation loss throughout the training process. Then the performance is evaluated in the testing dataset. For diverse analysis, we did our experiment for different groups of data. First, we analyze our experiments for our original 8-class labels. Then to analyze the range of the shots, we group our dataset into four different class labels(4-class) (three-point, two-point, mid-range, and free shot) and finally to analyze the scoring of the shots we analyze the model for two-class labels(2-class) (make/miss) of the shots.

Table 1: Accuracy comparison of different model for different class group

Methods	**Accuracy**					
	Subject Dependent			**Subject Independent**		
	8-class	**4-class**	**2-class**	**8-class**	**4-class**	**2-class**
3D ConvNet	59.48%	79.50%	77.31%	56.78%	77.42%	75.38%
3D optical ConvNet	51.72%	73.39%	72.47%	52.08%	74.44%	70.78%
Two stream 3D ConvNet	58.94%	74.79%	76.44%	51.85%	73.13%	76.06%

Table 1 shows the comparison of the accuracy for the 3D ConvNet for RGB data, optical flow data, and two-stream model with average fusion. We observe that among different input data in CNN models, the accuracy is mostly high with 3D ConvNet with RGB videos as compared to 3D ConvNet with optical video. The 3D ConvNet model has higher accuracy performance from RGB

videos than optical flow videos in all cases. The performance using two-stream networks has not improved the result significantly than using 3D ConvNet with RGB data. Also, among different groups of data, the model has higher accuracy in range classification(4-class) than in the 8-class group or 2-class group. Only that using two-stream 3D ConvNet the make and miss have shown higher performance compared to in 4-class in both subject dependent and subject independent classification than using single 3D ConvNet for RGB and optical dataset separately. We also observe that the accuracy is higher in most cases with subject-dependent analysis than subject independent analysis. Figures 4 and 5 present the confusion matrix based on recall and classification reports for the 4- class 3D ConvNet methods for subject dependent and subject independent classification methods. As described in section 5.1 and section 5.2, most of the misclassification is for mid-range which is mostly misclassified as a two-point or three-point shot. This can be due to the imbalanced nature of the dataset where there is a smaller number of mid-range data clips compared to 2-point and 3-point. Also, the identical similarity of the interclass activity and lack of specific range distinction between different ranges can be the reason for low performance. Also, we can see that the model can capture features of the free-throw which has significantly higher accuracy as compared to another group. During a free throw, the less movement of the players, constant camera angle, and low camera movement than in other activity groups can be the reason for the higher classification accuracy.

6. CONCLUSION AND FUTURE WORKS

In this study, we try to develop a dataset to classify the scoring action from a basketball game broadcasting video. Also, to learn the spatial and temporal features we used the state-of-the-art classification method of 3D ConvNet. Future work includes adding more activities of players in the game like dribbling, fouls, and scoring types for full automation of the activity recognition in the live videos. In most of the other dataset when deriving the optical flow dataset, the background will be static but in our case with the movement of the camera, results in the dynamic movement of the background. This has increased the challenges for improving the results using two-stream model and has also results in low classification performance with temporal information from optical data than considering only RGB spatial information.

Some of the other challenge it faces is the similar nature of the videos between different classes. The temporal information on the training will be not only from the movement of the ball or the player with the ball but it also tracks the movement of all the players on the court along with the referee and background audience. This dynamic nature of the dataset has added more challenges for classification. The experiments used for analyzing the performance on this dataset explained in this paper uses only basic parameters and features of the dataset. Analysis can be made by using higher dimensions input data with other deep learning networks considering the challenges the dataset presents. This can certainly help in increase the performance for classifying the scoring activities. This shows a higher possibility and opportunity to increase the performance using this dataset for further research.

REFERENCES

[1] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1704–1716, 2013.

[2] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, "Computer vision for sports: Current applications and research topics," *Comput. Vis. Image Underst.*, vol. 159, pp. 3–18, 2017.

[3] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A Hierarchical Deep Temporal Model for Group Activity Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB51: A Large Video Database for Human Motion Recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563.

[5] K. Soomro, A. R. Zamir, and M. Shah, "{UCF101:} {A} Dataset of 101 Human Actions Classes From Videos in The Wild," *CoRR*, vol. abs/1212.0, 2012.

[6] B. G. Fabian Caba Heilbron Victor Escorcia and J. C. Niebles, "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[7] W. Kay *et al.*, "The Kinetics Human Action Video Dataset," *CoRR*, vol. abs/1705.0, 2017.

[8] Yang Wang, Hao Jiang, M. S. Drew, Ze-Nian Li, and G. Mori, "Unsupervised Discovery of Action Classes," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2, pp. 1654–1661.

[9] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition," 2008.

[10] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification," in *Computer Vision -- ECCV 2010*, 2010, pp. 392–405.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," in *CVPR*, 2014.

[12] S. M. Safdarnejad, X. Liu, L. Udpa, B. Andrus, J. Wood, and D. Craven, "Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis," in *Proc. International Conference on Automatic Face and Gesture Recognition*, 2015.

[13] F. Chen, D. Delannay, and C. De Vleeschouwer, "An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study," *IEEE Trans. Multimed.*, vol. 13, no. 6, pp. 1381–1394, 2011.

[14] S. Su, J. Pyo Hong, J. Shi, and H. Soo Park, "Predicting Behaviors of Basketball Players From First Person Videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] G. Miao, G. Zhu, S. Jiang, C. Xu, and W. Gao, "A Real-Time Score Detection and Recognition Approach for Broadcast Basketball Video," 2007, pp. 1691–1694.

[16] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3264–3274, 2018.

[17] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[18] M. Abdullah, M. Ahmad, and D. Han, "Facial Expression Recognition in Videos: An CNN-LSTM based Model for Video Classification," in *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, 2020, pp. 1–3.

[19] J. You and J. Korhonen, "Attention Boosted Deep Networks For Video Classification," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1761–1765.

[20] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, "A 3D CNN-LSTM-based image-to-image foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 959–971, 2019.

[21] S. H. Bae, I. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 2016, pp. 11–15.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv Prepr. arXiv1406.2199*, 2014.

[23] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *European conference on computer vision*, 2016, pp. 744–759.

[24] Q. Xiong, J. Zhang, P. Wang, D. Liu, and R. X. Gao, "Transferable two-stream convolutional neural network for human action recognition," *J. Manuf. Syst.*, vol. 56, pp. 605–614, 2020.

[25] P. Thiam, H. A. Kestler, and F. Schwenker, "Two-stream attention network for pain recognition from video sequences," *Sensors*, vol. 20, no. 3, p. 839, 2020.

[26] R. Hetherington, "The Perception of the Visual World. By James J. Gibson. USA: Houghton Mifflin Company, 1950 (George Allen & Unwin, Ltd., London). Price 35s.," *J. Ment. Sci.*, vol. 98, no. 413, p. 717, 1952.

[27] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*, 2003, pp. 363–370.

[28] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[29] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2012.
[30] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European conference on computer vision*, 2010, pp. 140–153.
[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks." 2015.

AUTHORS

Sarbagya Ratna Shakya received the B. Eng. in Electronics Engineering from National College of Engineering, Tribhuvan University of Nepal in 2009; M. Eng. in Computer Engineering from Nepal College of Information Technology, Pokhara University of Nepal in 2014. Currently he is a PhD student in School of Computing Sciences and Computer Engineering, University of Southern Mississippi since 2016. His current research interests include machine learning, deep learning, and high-performance computing.

Chaoyang Zhang received his MS degree in computer science and PhD degree in computational analysis and modelling from Louisiana Tech University in 2001. He is currently a Professor of Computer Science in the School of Computing Sciences and Computer Engineering at the University of Southern Mississippi. He has published more than seventy papers in academic journals and conference proceedings. His research interests include data mining, machine learning, bioinformatics, image processing and high-performance computing.

Zhaoxian Zhou received the B. Eng. from the University of Science and Technology of China in 1991; M. Eng. from the National University of Singapore in 1999 and the PhD degree from the University of New Mexico in 2005. All His degrees are in Electrical Engineering. From 1991 to 1997, he was an electrical engineer in China Research Institute of Radio wave Propagation. He joined the University of Southern Mississippi in 2005. He has published more than fifty papers in academic journals and conference proceedings. His current research interests include computational science and electrical engineering.

NOVEL MACHINE LEARNING ALGORITHM FOR PREVALENT GENE BIOMARKERS FOR EFFECTIVE CANCER TREATMENT BY DETECTING ITS PH

Sahil Sudhakar Patil[1],Darshit Shetty[2],Vaibhav S. Pawar[*3,*4]

[1]Masters Student at Hof University of Applied Science
[2]MBA Student from Mumbai University, JBIMS
[*3]Associate Professor, Mechanical Engineering, Annasaheb Dange College of Engineering & Technology (ADCET), Ashta, Sangli, Maharashtra, India
[*4]PhD (Structures, IIT Bombay) (2013-2019), Graduated in August 2019

ABSTRACT

Patterns discovered from based on collected molecular profiles of patient tumour samples, and also clinical metadata, could be used to provide personalized cancer treatment to patients with similar molecular subtypes. Computational algorithms for cancer diagnosis, prognosis, and therapeutics that can recognize specific functions and aid in classifiers based on a plethora of publicly accessible cancer research outcomes are needed. Machine learning, a branch of artificial intelligence, has a great deal of potential for problem solving in cryptic cancer datasets, as per a literature study. We focus on the new state of machine learning applications in cancer research in this study, illustrating trends and analysing major accomplishments, roadblocks, and challenges along the way to clinic implementation. In the context of non-invasive treating cancer using diet-based and natural biomarkers, we propose a novel machine learning algorithm.

KEYWORDS

Biomarkers, Machine learning, Statistical Models, sequencing, pH sensing.

1. INTRODUCTION

There has been a continuous improvement in Cancer research over the past decades. Scientists used various methods, such as early-stage screening, to detect cancer types before they cause symptoms. They've also developed new strategies for predicting cancer treatment outcomes early on. Due to the introduction of new technologies a large amount of cancer data is available to the medical community. The accurate prediction of a disease outcome, on the other hand, is one of the most interesting and difficult tasks for physicians. As a result, medical researchers are increasingly using machine learning methods.

We present a study that use machine learning methods in cancer prediction and prognosis, in light of the growing trend of applying these methods to cancer prediction and prognosis. Prognostic and predictive features are considered in these studies, which may be independent of a specific treatment or are combined to guide cancer patient therapy [2]. Furthermore, we discuss the types

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 155-170, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110713

of machine learning methods used, the types of data they integrate, and the overall performance of each proposed scheme, as well as their benefits and drawbacks.

Integration of mixed data, such as clinical and genomic data, is a clear trend in the proposed works. However, we noticed a common problem in several works: the lack of external validation or testing of their models' predictive performance. It is clear that using machine learning methods to predict cancer susceptibility, recurrence, and survival could improve accuracy. According to [3,] the accuracy of cancer prediction outcome has improved by 15%–20% in recent years thanks to the use of machine learning techniques.

2. BIOMARKERS

Biomarkers (short for biological markers) are biological indicators of health. "A biomarker is defined as "an objectively measured and examined indicator of normal biochemical functions, pathogenic processes, or pharmacological reactions to a therapeutic treatment."Biomarkers are clinical measurements such as blood pressure or cholesterol levels that are used to monitor and predict health states in individuals and populations in order to plan appropriate therapeutic interventions. Biomarkers can be used individually or in combination to assess a person's health or disease state.

Today, a wide variety of biomarkers are used. Biomarkers are unique to each biological system (for example, the cardiovascular system, metabolic system, or immune system). Many of these biomarkers are simple to measure and are included in routine medical exams. A general health check, for example, might include measurements of blood pressure, heart rate, cholesterol, triglycerides, and fasting glucose. Weight, BMI, and waist-to-hip ratio are all common body measurements used to diagnose obesity and metabolic disorders. An ideal biomarker possesses certain characteristics that make it suitable for assessing a specific disease state. An ideal marker should have the following characteristics: Safe and simple to measure Follow-up is cost-effective. Treatment is modifiable; Gender and ethnic groups are all treated the same.

Biomarkers in disease principles have been applied to cancer detection, screening, diagnosis, treatment, and monitoring. Anti-cancer drugs used to be agents that killed both cancerous and healthy cells. More targeted therapies, on the other hand, have now been developed that can be directed to only kill cancer cells while leaving healthy cells alone. The evaluation of a common cancer biomarker aids in the development of therapies that target the biomarker. This can help to reduce the risk of toxicity while also lowering the cost of treatment. Genetic studies are important in cancer research because genetic abnormalities frequently underpin cancer development. As a result, specific DNA or RNA markers may aid in the detection and treatment of specific cancers.

2.1. Classification of Biomarkers

Biomarkers can be classified as follows:

- Type 0 biomarkers (natural history biomarkers): They aid in determining a disease's natural history and how it correlates with known clinical indicators over time.
- Type 1 biomarkers (drug activity biomarkers): These indicate the effect of drug intervention. They may be further divided[3] into:-

1. Efficacy biomarkers – describing a drug's therapeutic effects
2. Mechanism biomarkers – providing information on a drug's mechanism of action
3. Toxicity biomarkers – a symbol for a drug's toxicological effects

- Type 2 biomarkers (surrogate markers): They can be used to predict the effect of a therapeutic intervention and can also be used to replace a disease's clinical outcome.

Cytokine	Sample	Current purpose as a biomarker	Current well-known function in lung cancer
IL-6	Blood, BALF, and pleural effusion	Diagnostic, prognostic, and predicting the treatment response	Prooncogenic
IL-8	Blood, BALF, and sputum	Diagnostic and prognostic	Prooncogenic
VEGF	Blood, BALF, sputum, pleural effusion, and tissue	Diagnostic and prognostic	Prooncogenic
TNF-α	Blood	Predicting the treatment response	Prooncogenic and antitumor, depending on the context
IL-2	Blood	Prognostic and predicting the treatment response	Not yet determined
IL-18	Blood, BALF, and sputum	Diagnostic	Not yet determined
IL-10	Blood	Diagnostic	Prooncogenic and antitumor, depending on the context
IL-13	Blood	Diagnostic	Not yet determined
IL-22	Blood	Diagnostic and prognostic	Prooncogenic
IFN-γ	Blood	Diagnostic and prognostic	Not yet determined
IL-32	Tissue	Prognostic	Prooncogenic
IL-37	Tissue	Prognostic	Antitumor

Figure 1.Types of Biomarkers

3. MACHINE LEARNING

Machine learning is a branch of artificial intelligence characterized as a software program that can learn quickly by completing a set of tasks. There are three crucial aspects to consider. Describe how machine learning works. Tasks, experience, and performance are three of these factors. Tasks are datasets that are used to train the computer in order to improve its performance. With time and practice, the computer system can refine its model and predict the answer to a topic based on what it has learned from previous attempts. Machine learning employs a variety of algorithms, but they are divided into two categories: supervised and unsupervised learning. The supervised learning group includes any method that uses a set of training data. Each example has an input and output object in the dataset. The algorithm must work on manually entered answers in order to classify the result. This method of working is extremely reliant on the training data. As a result, the set must be correct for the algorithm to understand the data. The algorithm finds undetected patterns in a large amount of data in unsupervised learning. In this method, the computer algorithm is allowed to run and see what patterns will emerge as a result. As a result, there is no clear answer that can be considered correct or incorrect. There are dependent and independent variables in machine learning. The values that control the experiment are stored in the independent variables, which are also known as predictors or control input. The independent variables control the dependent variables, also known as output values.

3.1. Deep learning Architectures

Machine learning includes a subset called deep learning. It's a method of learning that works with multi-level layers and progresses to a more abstract level. The term "deep" refers to the multiple layers of nodes in a neural network. Based on the output from the previous layer variables, each layer in the network was trained on a different feature. The architecture of deep learning is based on neurons and is inspired by the layout of the human brain. A large number of neurons are connected in the human brain, forming a network of communication via signals received. An

artificial neural network is the name given to this idea (ANN). The algorithm in ANN creates layers that pass input values from one layer to the next, eventually resulting in a result.

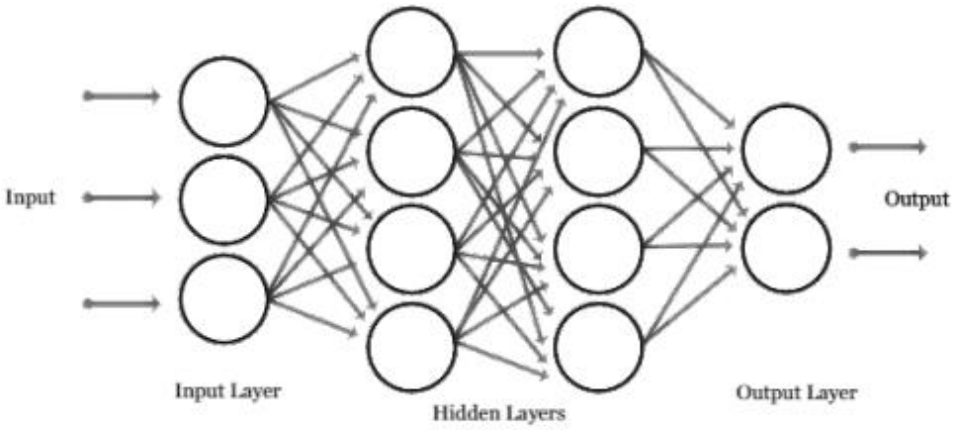


Figure 2.Architectures

Humans do not interfere with the layers of a neural network or the information being processed when using deep learning. Because the system algorithms are trained using data and learning procedures, it does not require human intervention. The method gains the ability to handle data with more dimensions.

3.2. Pre-processing of Genomic Data

Many algorithms can handle vector-matrix data, but converting DNA sequences to matrices is a different story. Values in genomic data are not supposed to be processed as standard text, so the data must be converted into a format suitable for the model. Label encoding and one-hot encoding, which converts nucleotide bases into numerical matrix form with 4-dimensional vectors, are used to accomplish this. Label Encoder converts the input into numerical labels between a value of 0 and N-1 using the Sklearn library ().The one-hot encode method avoids creating a hierarchy problem for the model with the label encode data by using Sklearn's one-hot encoding ( ) function. It changes the sequence by dividing the values into columns and converting them to binary numbers with only 0 and 1 values. This is done because the deep learning algorithm can't work with categorical data or words directly, so transforming input values makes the data more expressive and allows the algorithm to perform logical operations.

3.3. Pre-processing of image data

Image processing is a method of manipulating images to enhance or extract useful information from them so that an AI model can process them. The math function (x,y), where x and y are the image coordinates [40], defines an image as a two-dimensional array of numbers. Pixel values ranging from 0 to 255 are represented by the array numbers. The image height, colour scale, width, and number of levels/pixel are all image input parameters. Red, green, and blue (RGB) colour scales are also known as channels. The first step in pre-processing is to make sure that all of the images are the same size. Cropping the images allows you to change the size. The next step is to resize the photos once all of them have the same aspect ratio. Using a variety of library functions, they can be up scaled or downscaled. They're also normalized to ensure that the data distribution is consistent. The pixel values have been normalized to be between 0 and 1. This is because a network processes inputs using weight values, and smaller values can speed up the

learning process. The size of the image can also be reduced by converting the RGB channel into a grey scale image.

4. MACHINE LEARNING METHODS

The six machine learning techniques of K-nearest neighbour (KNN), Nave Bayes, Ada Boost, Support Vector Machine (SVM), Random Forest, and Neural Network with 10-cross fold technique were used to predict early lung tumours based on metabolomics biomarker features. The classification algorithm SVM aims to create a decision boundary between two categories that allows labels to be predicted from feature vectors [10]. When there is little prior knowledge of data, K-nearest-neighbour (KNN) is the preferred selection method, which is an elementary and straightforward nonparametric classification method [11]. The statistical classifier Nave Bayes was used to predict the probability of class membership [13]. It is hypothesized that all variables contribute independently to classification and that the outcome can be used to predict. [14] the goal of a neural network is to simulate the neuron and the human brain. The Neural Network's artificial neuron uses specific input features to assign mathematical weights that can eventually predict some output object.

4.1. K nearest Neighnors [KNN]

The algorithms for K Nearest Neighbors work. As per the data point's neighbor's similar characteristics. This algorithm used significant positive correlation to forecast the value of a new statistic and allocate the value depending as to how highly correlated we points in the training dataset were. It was used to determine whether or not the patient had cancer. This Algorithm is the best example of implementation.

KNN Is a Nonlinear Learning Algorithm

The ability of machine learning algorithms to estimate nonlinear relationships is a second feature that distinguishes them. Models that predict using lines or hyper planes are known as linear models. The model is shown as a line drawn between the points in the image. The linear model y = ax + b is the most well-known example. In the diagram below, you can see how a linear model could fit the example data.

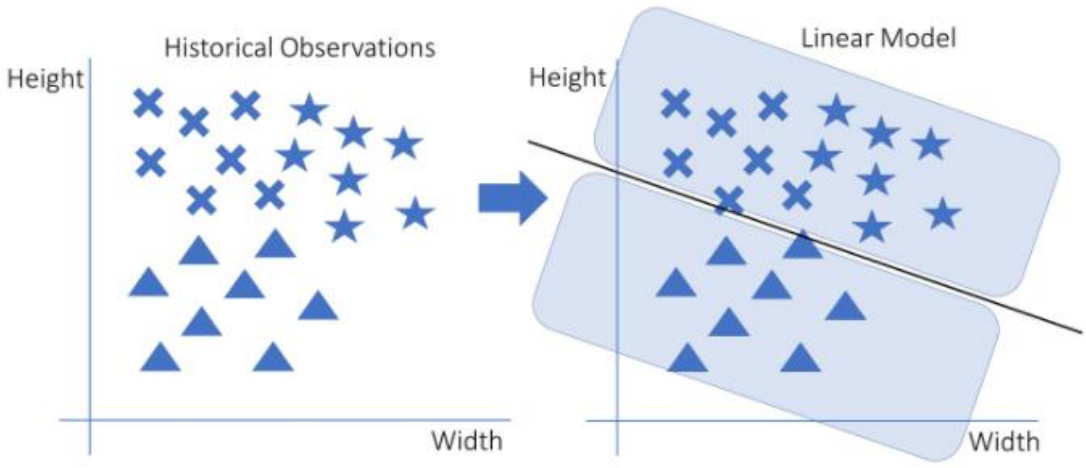


Figure 3. Linear Model

The data points are represented on the left with stars, triangles, and crosses in this illustration. A linear model on the right can distinguish triangles from non-triangles. Every non-triangle point is above the line, and every triangle point is below the line. If you wanted to add another independent variable to the previous graph, you'd have to draw it as a separate dimension, resulting in a cube with the shapes inside. A line, on the other hand, would not be able to split a cube into two parts. The hyper plane is the line's multidimensional counterpart. Nonlinear models are those that separate their cases using a method other than a line. The decision tree, which is essentially a long list of if... else statements, is a well-known example. If...else statements in the nonlinear graph would allow you to draw squares or any other shape you wanted. A nonlinear model is applied to the example data in the graph below.

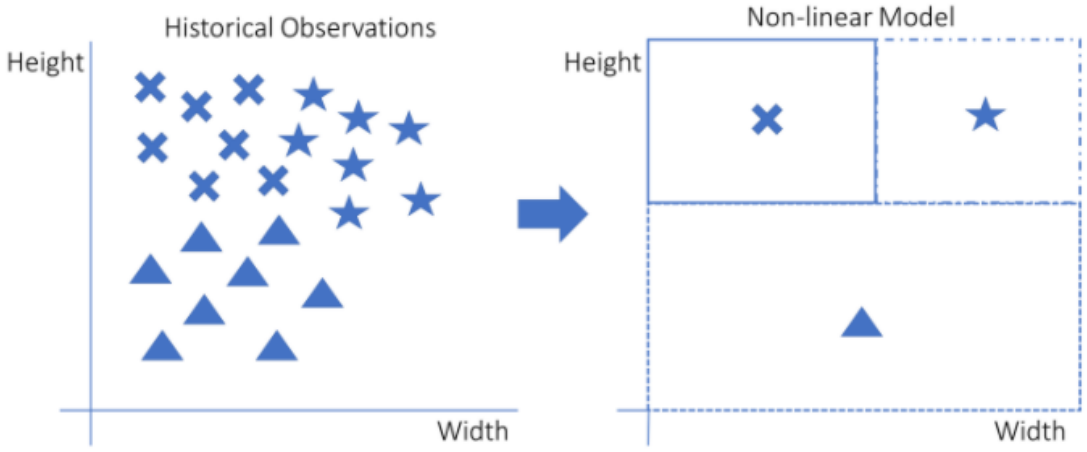


Figure 4. Non-Linear Model

4.2. Support Vector Machine [SVM]

Support Vector Machine (SVM) is a supervised learning classification algorithm broadly used in the development of cancer diagnosis and prognosis. The support vector machine algorithm's goal is to find a hyper plane in an N-dimensional space (N — the number of features) that categorizes data points clearly.

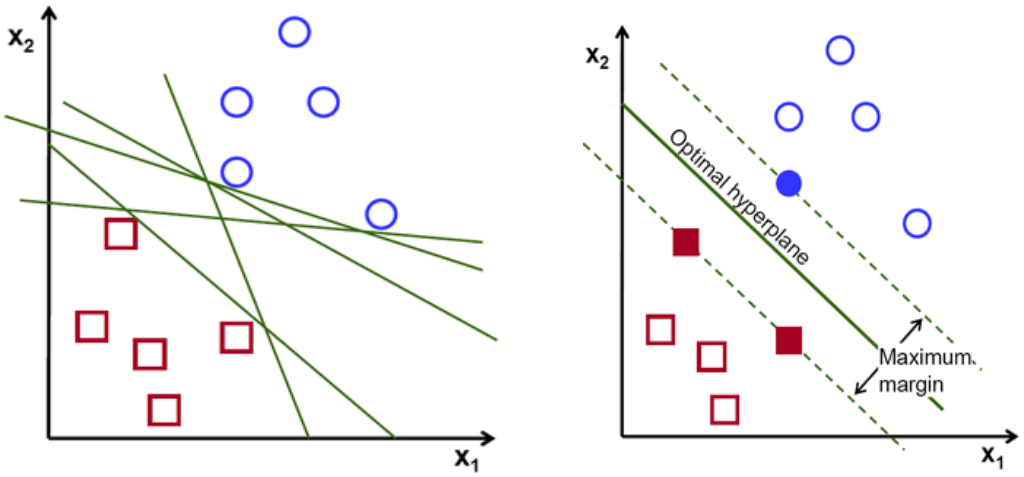


Figure 5. Possible Hyperplanes

There are numerous hyper planes from which to choose to separate the two classes of data points. Our goal is to find a plane with the greatest margin, or the greatest distance between data points from both classes. Trying to maximize the margin distance does provide some reinforcement, making it easier to classify future data points.

Hyper Planes and Support Vectors

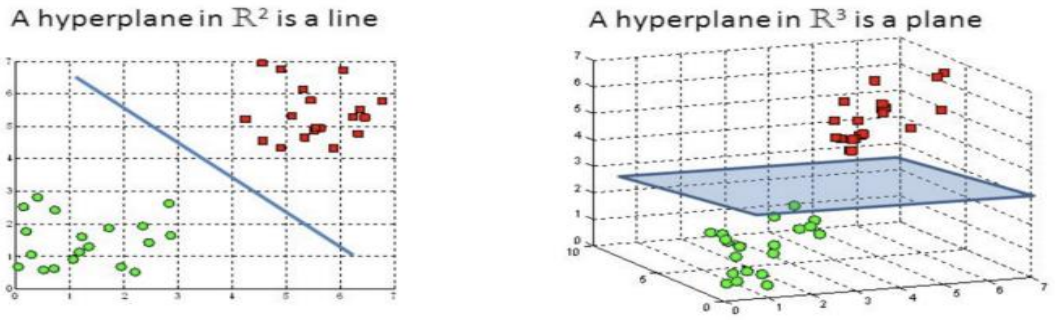


Figure 6. Hyperplanes in 2D and 3D feature space

Hyper planes are operating rules that aid in data classification. Different classes can be assigned to data points on each side of the hyper plane. The Hyper plane's dimension is also determined by the number of features. If there are only two input characteristics, the hyper - plane is just a line.

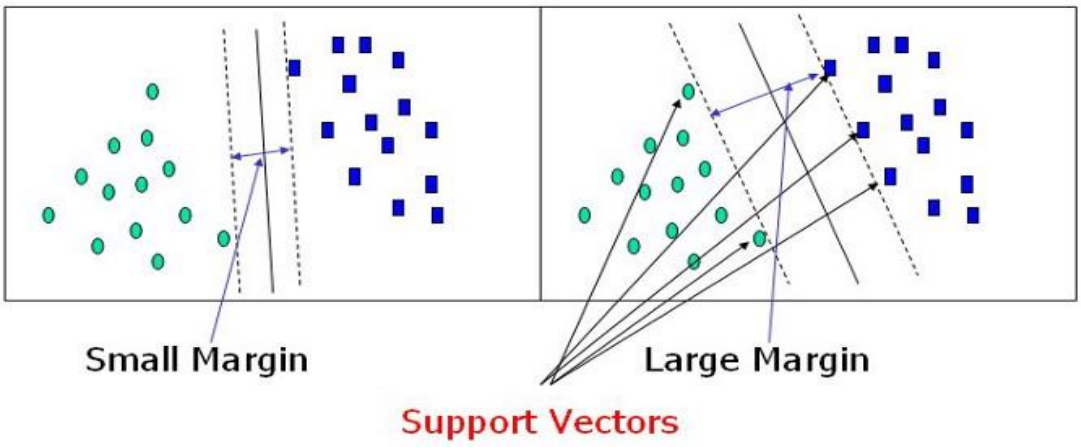


Figure 7. Support Vectors

Support vectors are data points that are nearer to the hyper plane and have an impact on the hyper plane's coordinate system. We significantly increase the classifier's margin by using these support vectors. The hyper plane's position will be altered if the support vectors are deleted. These are all the points that will assist us in constructing our SVM.

4.3. Naïve Bayes [NB]

Navie Bayes is a classification technique that relies on the Bayes theorem with independence among predictor variables. This particular feature in a class of features in a particular class is not

linked to the occurrence of any other features. If all these characteristics are reliant on each other, then these properties add value to the possibility of the class individually, which is the main reason for calling this “Naïve”. Navie Bayes is called "Nave" because it assumes that the characteristics of a measurement are independent of one another. Bayes is naive because he is almost never correct. It's simple to bold, and it's especially useful for large data sets. Naive Bayes is a sophisticated classification method that is known for its simplicity. The Naive Bayes model is simple to construct and is especially useful for large data sets. Naive Bayes is known to outperform even the most sophisticated classification methods due to its simplicity. The Bayes theorem allows you to calculate posterior probability P(c|x) from P(c), P(x), and P(x|c) using P(c), P(x), and P(x|c). Take a look at the following equation.

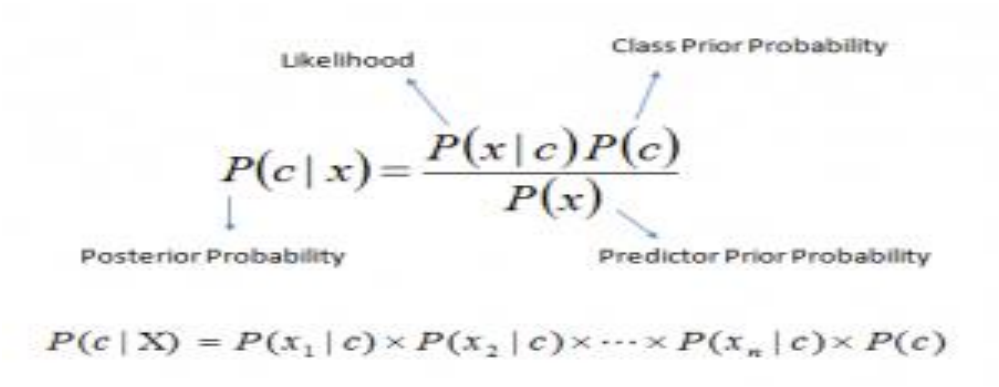


Figure 8. Bayes Theorem

4.4. Logistic Regression [LR]

It's a categorical dependent variable that's used in a classification algorithm. The goal of Logistic Regression is to discover a link between features and the likelihood of a specific outcome. Binomial Logistic Regression is a type of problem in which the response variable has two values: 0 and 1, or pass and fail, or true and false. When the response variable can have three or more possible values, Multinomial Logistic Regression is used. The basic components of a machine learning method are data samples. Every sample has a number of features, out of which each has a different set of values. Furthermore, knowing what type of data will be used ahead of time allows for proper tool and technique selection for their analysis. Some data-related issues concern the data's quality and the steps taken to prepare it for machine learning. Noise, outliers, missing or duplicate data, and biased-unrepresentative data are all examples of data quality issues. When data quality is improved, the quality of the resulting analysis is usually improved as well. Additionally, pre-processing steps focusing on data modification should be used to improve the raw data's suitability for further analysis. There are a variety of data pre-processing techniques and strategies that focus on modifying the data for better fit in a specific ML method. Some of the most important techniques are I dimensionality reduction, (ii) feature selection, and (iii) feature extraction. Dimensionality reduction has numerous advantages when datasets have a large number of features. When the dimensionality of the data is low, ML algorithms perform better [15]. Reduced dimensionality can also eliminate irrelevant features, reduce noise, and produce more robust learning models because fewer features are involved. The process of reducing dimensionality by selecting new features that are a subset of the old ones is known as feature selection. For feature selection, there are three main approaches: embedded, filter, and wrapper [15]. In the case of feature extraction, the initial set of features can be used to create a new set of features that captures all of the important information in a dataset. The creation of new sets of features allows the benefits of dimensionality reduction to be gathered.

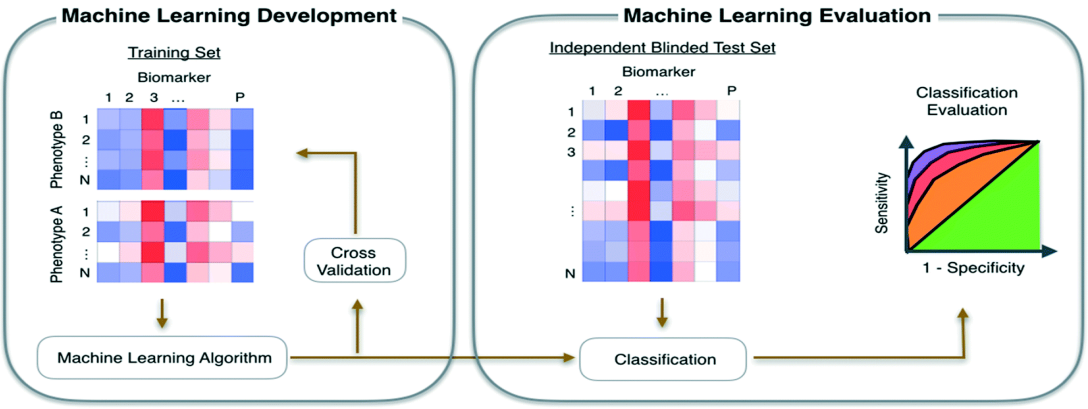


Figure 9.Machine Learning Evaluation

The use of feature selection techniques, on the other hand, may cause specific fluctuations in the creation of predictive feature lists. Several studies have been published that discuss the lack of agreement between different groups' predictive gene lists, the need for thousands of samples to achieve desired results, the dangers of information leaks and the lack of biological explanation of forecasting signatures.

The primary goal of machine learning techniques is to create a model that can be used for classification, prediction, estimation, or any other task. Classification is the most common task in the learning process. This learning function, as previously stated, classifies the data item into one of several predefined classes. Training and generalization errors can occur when ML techniques are used to create a classification model. The former refers to training data misclassification errors, while the latter refers to expected testing data errors. A good classification model should be able to accurately classify all of the instances in the training set. The phenomenon of model over fitting occurs when a model's test error rates begin to rise while its training error rates fall. This situation is related to model complexity, which means that as the model complexity increases, the training errors of the model will decrease. Obviously, the ideal complexity of a non-over fitting model is the one that produces the smallest generalization error. The bias–variance decomposition is a formal method for analysing a learning algorithm's expected generalization error. The error rate of a learning algorithm is measured by the bias component of that algorithm. Variation in the learning method is a second source of error that affects all possible training sets and test sets of a given size. The sum of bias and variance, referred to as the bias–variance decomposition, is the overall expected error of a classification model.

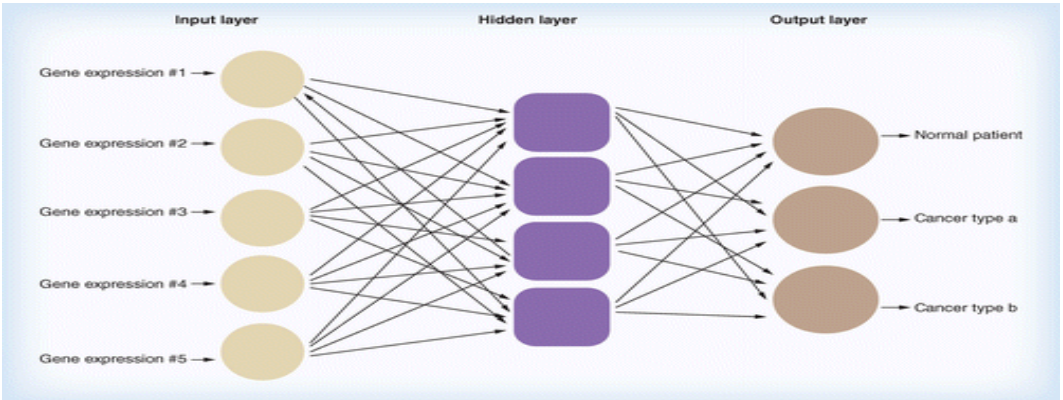


Figure 10. Classification of Gene Expression

5. PATIENT DATA

- Image
- What is age
- Previous background of illness /heredity
- Previous medication for disease
- Immediate history
- Whether any disorder since long
- Significant change in diet
- Different habits such as smoking
- Typical symptoms from start
- Severity of symptoms
- Medication once symptoms initiated
- Symptoms improved or as they were
- Which medical treatment doctor as prescribed
- After medication new symptoms

6. MACHINE LEARNING APPROACH BY DETECTING ITS PH

Cancerous cells differ from healthy cells in several ways that help distinguish them as dangerous. For example, the pH (acidity level) of a cancerous cell differs from the pH of a healthy cell.

Over the last few decades, immunofluorescence has been widely used in a variety of biological and biomedical applications to visualize specific biological phenomena at the cellular and subcellular levels. Despite the fact that it has a number of disadvantages To begin with, fluorophores have the ability to Phototoxic effects are caused by the generation of reactive oxygen species, which have been shown to have harmful effects, negative consequences for cell physiology and health.

Phototoxic damage can be measured and reduced, but it cannot be eliminated. Furthermore, because antibodies cannot cross the cell membrane, immunofluorescence necessitates a cell fixation step. As a result, any downstream analysis that necessitates the presence of living cells is no longer possible. In addition, some research areas, such as in vitro stem cell and drug discovery studies, require very little cell manipulation. To enable scientists to extract valuable information from living cells, new efficient and sensitive alternative methods are required. Intracellular acidity has been shown to be a useful tool for studying single cells, among other things. Intracellular acidity, in particular, is linked to a variety of physiological processes, including cell migration, division and apoptosis and influences how the entire cellular environment functions by regulating events ranging from enzymatic activity to cytoskeletal structure dynamics. Ten to twelve Physiological pH ranges from 4.7 to 8.0 and abnormal intracellular acidity has been linked to the development of diseases like Alzheimer's and even heat stroke.

White and colleagues recently highlighted the importance of deregulated pH dynamics in cancer initiation, progression, and adaptation. In cancer cells, the intracellular pH is higher than in normal cells, while the extracellular pH is lower. This phenomenon has been seen in the early stages of cancer, with pH differences between intracellular and extracellular environments increasing as the cancer progresses. Increased intracellular pH has been linked to the epithelial-to-mesenchyme transition, which is linked to the initiation of metastatic disease.

To study cellular pH, a variety of methods have been developed, most of which rely on fluorescence indicators and decorated nanoparticles. However, they have drawbacks, such as

complex multi-step protocols for nanoparticle synthesis and functionalization. Photo bleaching, which is known to affect cell physiology, is also a factor that affects fluorescence imaging methods. In 2017, Hou et al. published the first paper on a novel single-cell pH-based imaging method, in which the authors were able to rapidly identify cancer cells using a combination of UV-vis micro-spectroscopy and common pH indicators. Innovative approaches to extracting valuable information from biological and medical images have been enabled by numerous advancements in the field of computer vision. Specifically, ML-based algorithms have been developed to extract multiple features from single cells and even subcellular components, which can then be used to identify complex phenotypes and diagnose diseases.

7. METHODOLOGY

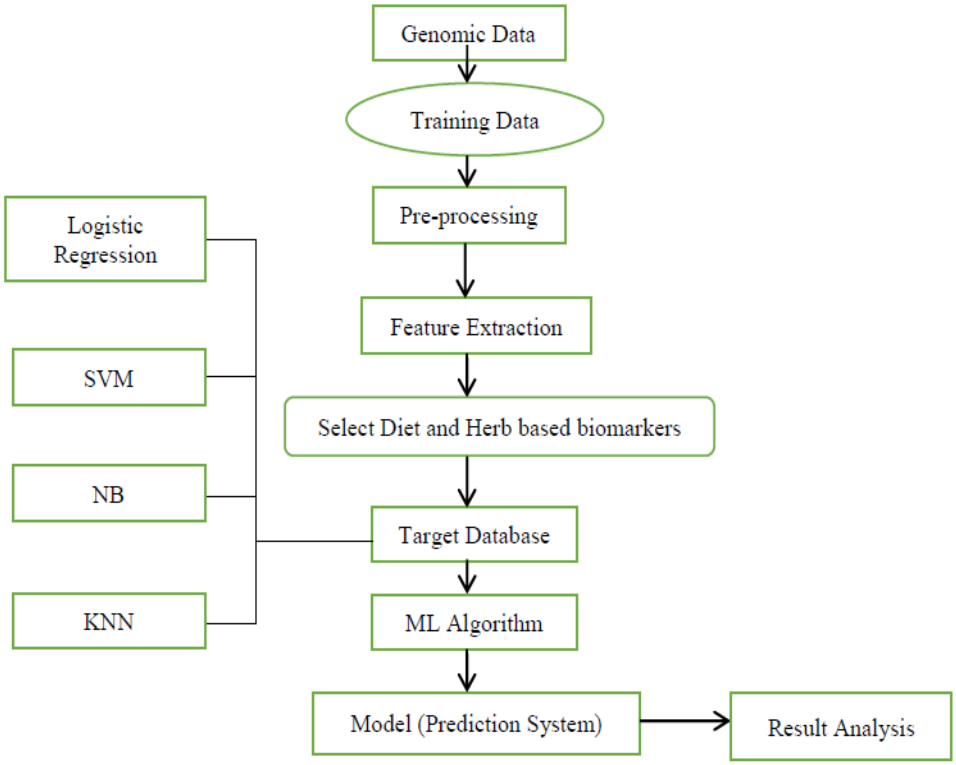


Figure 11. Methodology

8. ALGORITHM STEPS

- Defining the problem and fixing the parameters highly robust and flexible machine learning model
- Deciding severity index and defining mathematical preliminaries
- Assembling of data set
- Choosing a measure of success
- Deciding on an evaluation protocol
- Preparing the data
- Data sorting and cleaning
- Developing a model that does better than a baseline
- Developing a model that over fits and regularizing the model and tuning the parameters

9. CONCLUSIONS

In this study, we looked at machine learning concepts and how they can be used to predict and prognosis cancer. The majority of recent studies have centred on the development of predictive models using supervised machine learning methods and classification algorithms with the goal of accurately predicting disease outcomes. Based on their findings, it's clear that combining multidimensional heterogeneous data with various feature selection and classification techniques can result in promising inference tools in the cancer domain.

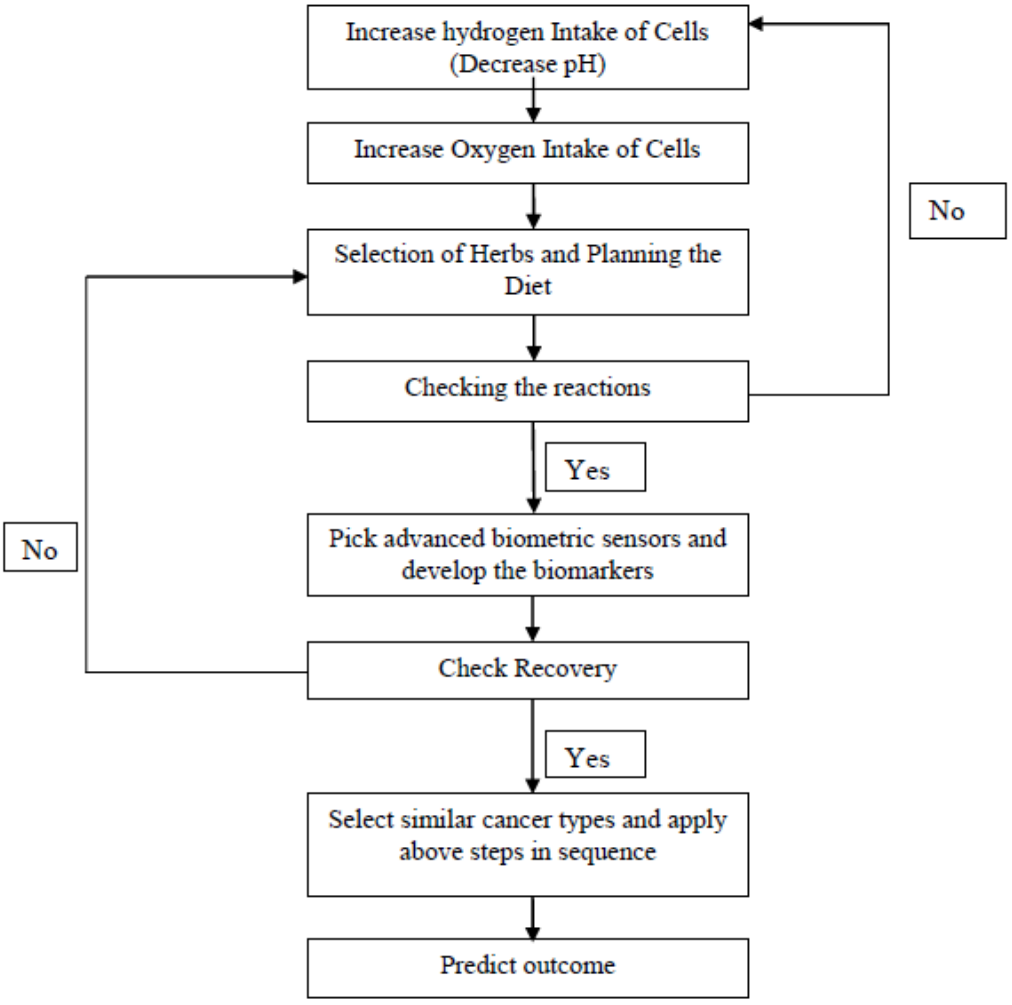


Figure 12.Flowchart for Machine Learning Algorithm

We proposed novel machine learning algorithm as portrayed in the Flowchart above. Idea that is being executed is based upon herb based diagnostics for cancer treatment. Accordingly, deep learning from the infected cells, multi-regression based machine learning model is being developed. Simple biomarkers with non-invasive treatment are being attempted along with appropriate genome sequencing.

10. INITIAL RESULTS

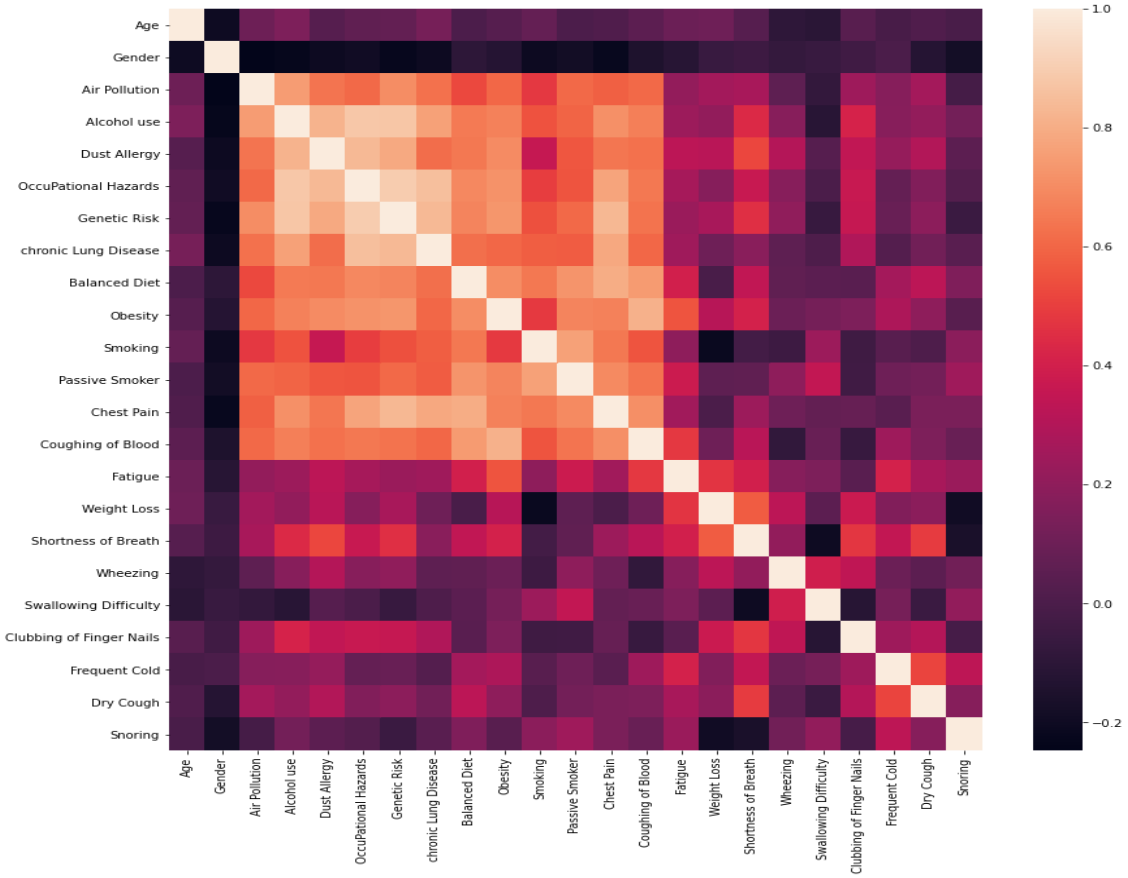


Figure 13. Visual Understanding of Correlation between Patient Characteristics

Data was taken from the publicly available data world for the initial observation of the results. Their sociodemographic characteristics, lifestyle characteristics, and external and internal residential area characteristics were all included in the data. Based on data collected, regression models were created to investigate the links between lung cancer and urban spatial factors.

Individual level factors serve as control variables, attempting to capture the socioeconomic status and lifestyle of surveyed residents, both of which have an impact on their health outcomes. Age, gender, workplace, tobacco use (smoking history and family/colleague smoking status), cooking fume exposure, duration of outdoor exercise, and chronic medical history are all factors to consider.

The findings back up the hypothesis that both indoor and outdoor spatial factors are linked to lung cancer incidence. To revise the criteria for lung cancer screening of high-risk individuals, certain principles based on modeling results are proposed. This will help in the further study to investigate the herbs necessary for the treatment, that will help in maintaining the pH level.

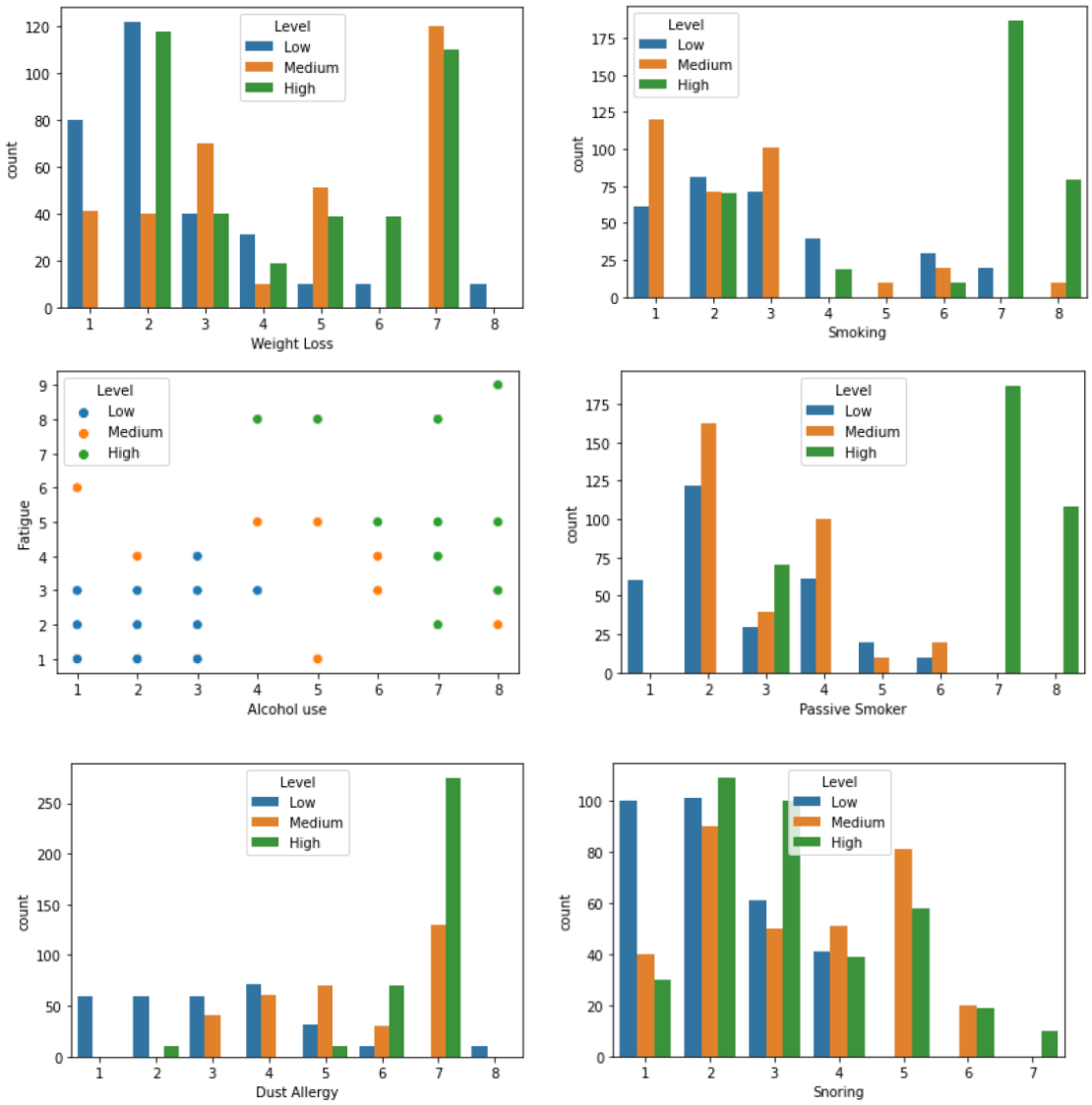


Figure 14. Classification on the basis of lifestyle Characteristics

- Following results were also obtained for various lifestyle characteristics which were classified into low, medium and high level.
- Under Smoking the bar chart illustrates the number of low, medium and high levels of smokers which are divided into 8 groups. It can be seen that from group 1 to 4 there is an undulated trend between low and medium smokers. A sudden increase is seen in high level smokers from group 6 to 7. A fall in medium level smokers is seen in group 5. Overall there is a fluctuating trend in all groups except for group 7 and 8.
- Under Dust allergy the bar chart illustrates the number of low, medium and high levels of people with dust allergy which are divided into 8 groups. A constant trend of low level allergic people is seen between groups 1- 3. There is rise in people with medium level dust allergy between groups 3 - 5. Though only few people with low dust allergy are seen in group 8. A sudden increase in group 7 with high level allergic people is striking. Overall low level allergic people are less as compared to high and medium levels.

- Under Snoring the bar chart illustrates the number of low, medium and high levels of people with snoring disorder which are divided into 8 groups. A constant decrease in low and high level of snoring disorder is seen between group 1-4 and 2-4 respectively. There is a dip in medium level people from group 2 - 5. Overall group 5, 6 and 7 have nil low level people and least high level people with snoring disorder.
- Overall it is observed that, each parameter has fluctuating effect over the patient number count. It is indeed essential to have the combine effect of all the parameters via multi-regression study. Furthermore, diet based and herb based studies need to be attempted in the context of pH and overall health and outcome of Cancer patients. These things are being accommodated in the proposed model suggesting the importance of Novel Machine learning algorithm with herb and diet based biomarkers along with the existing scheme.

REFERENCES

[1] D. Hanahan, R.A. Weinberg, Hallmarks of Cancer: the next generation Cell, 144(2011), pp. 646-674
[2] M.Y.C. Polley, B. Freidlin, E.L. Korn, B.A. Conley, J.S. Abrams, L.M. Mc hane, Statistical and practical considerations for clinical evaluation of predictive biomarkers, J Natl Cancer Inst, 105 (2013), pp. 1677-1683.
[3] J.A. Cruz, D.S. WishartApplications of machine learning in cancer prediction and prognosis, Cancer Informat, 2 (2006), p. 59
[4] O. Fortunato, M. Boeri, C. Verri, D. Conte, M. Mensah, P. Suatoni, et al. Assessment of circulating microRNAs in plasma of lung cancer patients, Molecules, 19 (2014), pp. 3038-3054
[5] H.M. Heneghan, N. Miller, M.J. Kerin, MiRNAs as biomarkers and therapeutic targets in cancer, CurrOpinPharmacol, 10 (2010), pp. 543-550
[6] D. Madhavan, K. Cuk, B. Burwinkel, R. Yang, Cancer diagnosis and prognosis decoded by blood-based circulating microRNA signatures, Front Genet, 4 (2013)
[7] K. Zen, C.Y. Zhang, Circulating microRNAs: a novel class of biomarkers to diagnose and monitor human cancers, Med Res Rev, 32 (2012), pp. 326-348
[8] S. Koscielny, Why most gene expression signatures of tumors have not been useful in the clinic, SciTransl Med, 2 (2010) [14 ps12-14 ps12]
[9] S. Michiels, S. Koscielny, C. Hill, Prediction of cancer outcome with microarrays: a multiple random validation strategy, Lancet, 365 (2005), pp. 488-492
[10] Y. Sun, S. Goodison, J. Li, L. Liu, W. Farmerie, Improved breast cancer prognosis through the combination of clinical and genetic markers, Bioinformatics, 23 (2007), pp. 30-37
[11] L. Bottaci, P.J. Drew, J.E. Hartley, M.B. Hadfield, R. Farouk, P.WR. Lee, et al., Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions, Lancet, 350 (1997), pp. 469-472
[12] P.S. Maclin, J. Dempsey, J. Brooks, J. Rand, Using neural networks to diagnose cancer, J Med Syst, 15 (1991), pp. 11-19
[13] R.J. Simes, Treatment selection for cancer patients: application of statistical decision theory to the treatment of advanced ovarian cancer, J Chronic Dis, 38 (1985), pp. 171-186
[14] M.F. Akay, Support vector machines combined with feature selection for breast cancer diagnosis, Expert SystAppl, 36 (2009), pp. 3240-3247
[15] T. Pang-Ning, M. Steinbach, V. Kumar, Introduction to data mining (2006)

AUTHORS

Sahil Sudhakar Patil Bachelors in Mechanical Engineering 2012- 2016, Pursuing Master in Operational excellence at Hof University of Applied Science (2020-2022)

Darshit Shetty Bachelors in Mechanical Engineering 2008-2012; Pursuing Masters in Marketing Management from Mumbai University, JBIMS.

Dr. Vaibhav S. Pawar*, Associate Professor, Mechanical Engineering, Annasaheb Dange College of Engineering & Technology (ADCET), Ashta, Sangli, Maharashtra, India; PhD (Structures, IIT Bombay) (2013-2019), Graduated in August 2019

DERIVING AUTISM SPECTRUM DISORDER FUNCTIONAL NETWORKS FROM RS-FMRI DATA USING GROUP ICA AND DICTIONARY LEARNING

Xin Yang[1], Ning Zhang[2] and Donglin Wang[3]

[1]Department of Computer Science, Middle Tennessee State University, Murfreesboro, TN, USA
[2]Department of Computer Information Sciences, St. Ambrose University, Davenport, USA
[3]Department of Mathematical Sciences, Middle Tennessee State University, Murfreesboro, TN, USA

ABSTRACT

The objective of this study is to derive functional networks for the autism spectrum disorder (ASD) population using the group ICA and dictionary learning model together and to classify ASD and typically developing (TD) participants using the functional connectivity calculated from the derived functional networks. In our experiments, the ASD functional networks were derived from resting-state functional magnetic resonance imaging (rs-fMRI) data. We downloaded a total of 120 training samples, including 58 ASD and 62 TD participants, which were obtained from the public repository: Autism Brain Imaging Data Exchange I (ABIDE I). Our methodology and results have five main parts. First, we utilize a group ICA model to extract functional networks from the ASD group and rank the top 20 regions of interest (ROIs). Second, we utilize a dictionary learning model to extract functional networks from the ASD group and rank the top 20 ROIs. Third, we merged the 40 selected ROIs from the two models together as the ASD functional networks. Fourth, we generate three corresponding masks based on the 20 selected ROIs from group ICA, the 20 ROIs selected from dictionary learning, and the 40 combined ROIs selected from both. Finally, we extract ROIs for all training samples using the above three masks, and the calculated functional connectivity was used as features for ASD and TD classification. The classification results showed that the functional networks derived from ICA and dictionary learning together outperform those derived from a single ICA model or a single dictionary learning model.

KEYWORDS

Functional connectivity, rs-fMRI, autism spectrum disorder (ASD), group ICA, Dictionary Learning.

1. INTRODUCTION

The human brain is very mysterious to humans since it is the most complex organ known in the world. Although attempts have been made for centuries to study and unravel the mystery of the human brain, our understanding of it is still limited. Therefore, a deep understanding of the brain is essential if humans are to unravel the relationship between brain function in neurological function-related disease. Functional magnetic resonance imaging (fMRI) is a technique that uses

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 171-184, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110714

magnetic resonance imaging to detect brain activity by calculating the fluctuations in local blood oxygenation level, which provides a means of understanding how spatially distributed brain regions interact and work together to create neurological function.

Prior to the development of functional neuroimaging methods, a series of developments and advances were experienced. In the 1970s, Allan M. Cormack and Godfrey N. Hounsfield invented the computer-assisted tomography imaging technique. This technique allows the acquisition of higher resolution images of brain structures. Soon after, the invention of radioligands led to two new neuroimaging techniques: single photon emission computed tomography (SPECT) and positron emission tomography (PET). MRI is a relatively new medical imaging technique. In the early years of the 21st century, developments in neuroimaging began to allow functional neuroimaging techniques. Compared to ionizing radiation methods such as computed tomography (CT)and positron emission tomography (PET), fMRI offers a completely safe and non-invasive method of imaging brain activity with reasonable spatial and temporal resolution. With the rapid development of functional magnetic resonance imaging (fMRI), modern cognitive neuroscientists now have an imaging tool that overcomes the limitations of earlier neurocognition studies. Since then, the field of fMRI has led to remarkable progress in biomedical research [1].

Researchers were most initially interested in fMRI for the brain's response to external mental stimulation. Thus, most of the initial studies focused on the response to external task-evoked activities [2]. In 1995, Biswal et al [3,4] found that the primary motor cortex's left and right hemispheric regions were not in a silent state at rest. They were the first to hypothesize that the functional connectivity exhibited in the motor cortex is a general phenomenon and not due to external stimulus events. Their discoveries suggested that resting-state fMRI can also provide meaningful information on neurological function even in the absence of external events stimulation. Since then, there has been an explosion of subsequent studies of brain function using resting-state fMRI (rs-fMRI) data [5,6,7]. Scientists have come to recognize the usefulness of studying whether patterns identified in resting-state fMRI data exhibit the same characteristics under different conditions. Nowadays, rs-fMRI has become an essential technique for analyzing neurological disorders such as Alzheimer's disease, autism spectrum disease, etc.

Autism spectrum disorder (ASD) is a neurologically impaired brain disorder in which individuals have impaired development of social interaction and communication skills [8]. According to Centers for Disease Control (CDC), to date, approximately one in every 54 children has been diagnosed with an autism spectrum disorder in the United States. Research in autism spectrum disorder is imperative as more and more families around the world are affected by the disorder. The diagnosis of autism spectrum disorder is challenging and complex. The clinical approach to diagnosis is generally based on a comprehensive behavioral assessment by a child psychiatrist or psychologist, which includes observation of the child's behavior, speech and language, hearing, vision, motor function, etc. [8] The clinical approach has shed light on many aspects of autism spectrum disorder in behavior perspective.

Although brain neurologists and neuroscientists have suggested many causes, including genetic and environmental factors, the exact etiology of autism spectrum disorder remains unclear. In addition to the behavioral assessment, recent advances in neuroimaging techniques have prompted the possibility of interpreting the connectivity between behavioral disorders and neurological function from fMRI data. An increasing number of research demonstrates that social and communicative deficits are associated with the function and connectivity of cortical networks.

The theory of cortical under-connectivity has been proposed as an explanatory model for ASD, suggesting that abnormal functional connectivity between brain regions may contribute to poor performance on cognitive and social tasks in people with ASD [9, 10,11,12].

To explore the difference in brain connectivity between ASD and TD groups. In this study, we aim to compare functional connectivity in ASD groups and TD groups. We combined group independent component analysis (ICA) and dictionary learning together to identify functional networks and investigate their connectivity. ICA is a data-driven method that separates a multivariate signal into additive subcomponents [13]. ICA attempts to decompose a multivariate signal into a linear combinations of independent non-Gaussian signals. When ICA is applied to fMRI data, the 4D fMRI time series signal are typically modeled as linear combinations of unknown spatially independent activity patterns. The 4D fMRI time series signals are decomposed into spatially independent components (ICs), but temporally coherent networks. Spatial ICA has been applied to resting-state fMRI of anesthetized child patients by Kiviniemi. By analyzing the statistical characteristics of the observed data samples and minimizing the mutual information between the observed signals, ICA can separate out different source signals [14]. The problem arises, however, that ICA is required to comply with the orthogonality constraints on the data representation subspace, which leads to the fact that the maximum number of causes is often limited to the signal dimension. In response to this situation, this problem has triggered the emergence of a new promising research area, namely dictionary learning. The focus of Dictionary learning is to construct a dictionary of atoms or subspaces that provides efficient representations for the observed samples [14]. Dictionary learning has the potential to derive the priori unknown statistics for sparse signals. It has been successfully applied in the field of medical imaging, such as electroencephalogram (EEG), magnetic resonance imaging (MRI), and functional MRI (fMRI).

The objective of this study is to derive functional networks from autism spectrum disorder (ASD) groups using group ICA and Dictionary Learning and use the functional connectivity calculated from the derived functional networks to classify ASD and TD groups. In our experiments, the ASD functional networks were derived from resting-state functional magnetic resonance imaging (rs-fMRI) data. We downloaded a total of 120 training samples including 58 ASD and 62 TD participants, which were obtained from the Autism Brain Imaging Data Exchange I (ABIDE I) repository. The functional connectivity matrix was calculated from the derived functional networks, which have been applied as the classification features. Our methodology and results have five main parts. First, we utilize the group ICA model to extract ASD functional networks and rank the top 20 ROIs. Second, we utilize a dictionary learning model to extract ASD functional networks and rank the top 20 ROIs. Third, we merged the 40 selected ROIs from the two models together as the ASD functional networks. Fourth, we generate three corresponding masks based on the 20 selected ROIs from group ICA, the 20 ROIs selected from dictionary learning, and the 40 combined ROIs selected from both. Finally, we extract ROIs for all training sample using the above three masks, and the calculated functional connectivity was used to classify ASD and TD participants. The classification results showed that the functional networks derived from ICA and dictionary learning together outperform those derived from a single ICA model or a single dictionary learning model.

2. METHODS

2.1. Datasets

A total of 120 participants, 58 with a diagnosis of ASD and 62 TD participants, were included in this study. All data were obtained from the public repository: Autism Brain Imaging Exchange I

(ABIDE I). The ABIDE I represents the first ABIDE initiative. The ABIDE I datasets consists of structural MRI and resting-state fMRI data, and the corresponding phenotypic information. The fMRI data from ABIDE I were pre-processed using Configurable Pipeline for the Analysis of Connectomes (CPAC). CPAC is an open-source pipeline to pre-process resting-state fMRI data.

In these 120 subjects, there are 58 ASD and 62 TD subjects, of which 13 females and 107 males. The summary information of the selected 120 subjects is displayed in Table I. Table I contains a summary of phenotypic information for ASD and TD, such as sex, age, and experimental site name.

Table 1: ABIDE data phenotypical information summary

Site	Count		Count		Total	Age Range
	ASD	TD	M	F		
OHSU	12	13	25	0	25	8~15
OLIN	14	14	23	5	28	10~24
PITT	24	26	43	7	50	9~35
SDSU	8	9	16	1	17	12~17
TOTAL	58	62	107	13	120	8~35

In this study, we aim to combine group ICA and dictionary learning together to identify functional networks and investigate ASD and TD classification using functional connectivity calculated from the derived functional networks.

2.2. Spatial Independent Component Analysis on single-subject fMRI data

ICA is a blind source separation method for separating data into underlying informational components [14]. ICA assumes that the observed data samples can be decomposed into linear combinations of unknown underlying signals and that the data can be reconstructed based on statistical independence. Accordingly, ICA separates signal mixtures into statistically independent signals. ICA has been applied to different domains such as speech processing, neuroimaging (fMRI, EEG), telecommunications, and stock market prediction [15].

The ICA model is a statistical model with linear combinations of mixed signals. When we use the data matrix $\boldsymbol{X}$ to represent the observed data, the model is generally expressed in matrix form as following:

$$\boldsymbol{X} = \boldsymbol{AS}$$

where $\boldsymbol{X} = (\boldsymbol{X_1}, \dots, \boldsymbol{X_m})^{\boldsymbol{T}}$ is the observed data matrix with dimension $\boldsymbol{T} \times \boldsymbol{M}$, $\boldsymbol{A} = \left(\boldsymbol{a_{ij}}\right)$ is the unknown mixing matrix of size $\boldsymbol{T} \times \boldsymbol{K}$, and $\boldsymbol{S} = (\boldsymbol{S_1}, \dots, \boldsymbol{S_m})^{\boldsymbol{T}}$ is the $\boldsymbol{m}$ unknown source signals need to be recovered. In total, there are $\boldsymbol{K}$ sources. In the source matrix $\boldsymbol{S}$, each row $\boldsymbol{s_k^T}$ represents an independent component. In the mixing matrix $\boldsymbol{A}$, each column $\boldsymbol{a_k}$ represents its corresponding weights. This can be written in the form of linear weighted sums: $x_t = a_{t1}s_1 + \cdots + a_{tk}s_k$.

It's well-known that ICA algorithms are generally widely used for time series signals. However, it must be emphasized that the fMRI signals are time series of the spatial volumes, and the different activation patterns derived from the fMRI signals are also spatial-oriented. Therefore, spatial ICA is more suitable for fMRI data analysis [16]. The spatial ICA model is illustrated in Figure 1:

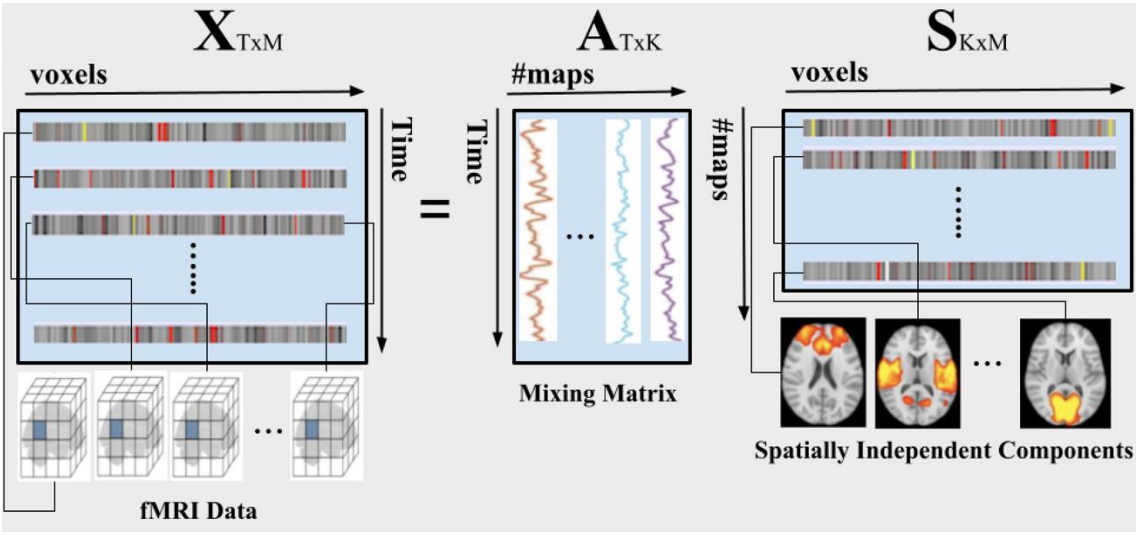


Figure 1: Spatial ICA for single fMRI data

From Figure 1 we can see that each row in the data matrix $\boldsymbol{X}$ represents a volume vector, and each row in the source matrix $\boldsymbol{S}$ represents an independent spatial pattern. In the mixing matrix $\boldsymbol{A}$, each column represents the activation time series. The $\boldsymbol{T} \times \boldsymbol{M}$ data matrix $\boldsymbol{X}$ represents the observed fMRI data. Here, $\boldsymbol{M}$ is the total number of voxels in a subject's brain and $\boldsymbol{T}$ is the number of fMRI time series points.

2.3. group ICA on multi-subject fMRI data

In the context of the prevalence of group analysis, a large number of studies on rs-fMRI have shown that the correlation patterns found in the BOLD signals are highly reproducible across populations [20]. Meanwhile, some of the network patterns derived from rs-fMRI by the ICA model are consistent at the group level. However, the ICA model can be sensitive to data variations, even just mild variations. Based on these factors, a direct comparison of patterns estimated from different individual subject is not meaningful. Instead, it would be more meaningful and reasonable to analyze group-level patterns specifically for each subject. For the group-level extraction of ICA patterns, researchers have adopted different strategies. The data volumes from Individual subjects can be concatenated together in a time series, and then the ICA model can be applied to the group data [22]. Beckmann and Smith [23] proposed a novel model that refers to tensorial extension of ICA, which will estimate the patterns across subjects in the same time course. However, these methods can not directly detect the difference between groups in terms of individual ICs. In order to be able to represent the variability due to individual differences between subjects, Varoquaux etc. proposed an updated group model, called CanICA, to extract group-level IC components. The advantage of their proposed model is that by using generalized canonical correlation analysis (CCA) it can identify a subspace of reproducible components across subjects [20].

The observed time series fMRI data for each subject $\boldsymbol{Y_s}$ can be consist of a set of independent spatial patterns $\boldsymbol{P_s}$ with observation noise. For each subject $\boldsymbol{Y_s}$, it takes the form:

$$\boldsymbol{Y_s = W_s P_s + E_s}$$

, where $\boldsymbol{W_s}$ is a loading matrix, and $\boldsymbol{E_s}$is the observation noise. Each subject $\boldsymbol{Y_s}$ activity can be described by subject-specific spatial patterns $\boldsymbol{P_s}$, which are a combination of the group-level patterns B and additional subject-variability, the $\boldsymbol{P_s}$ takes the form:

$$\boldsymbol{P_s = A_s B + R_s}$$

The group form can be written with vertically concatenated matrices: $\boldsymbol{P} = \{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots, \boldsymbol{P_s}\}$, $\boldsymbol{R} = \{\boldsymbol{R_1}, \boldsymbol{R_2}, \dots, \boldsymbol{R_s}\}$, and $\boldsymbol{A} = \{\boldsymbol{A_1}, \boldsymbol{A_2}, \dots, \boldsymbol{A_s}\}$, $s = 1, \dots, N$. The canICA model is illustrated in Figure 2. The following Figure describes the steps starting from individual fMRI subjects, to obtain the group-level independent components.

$$\boldsymbol{P = AB + R}$$

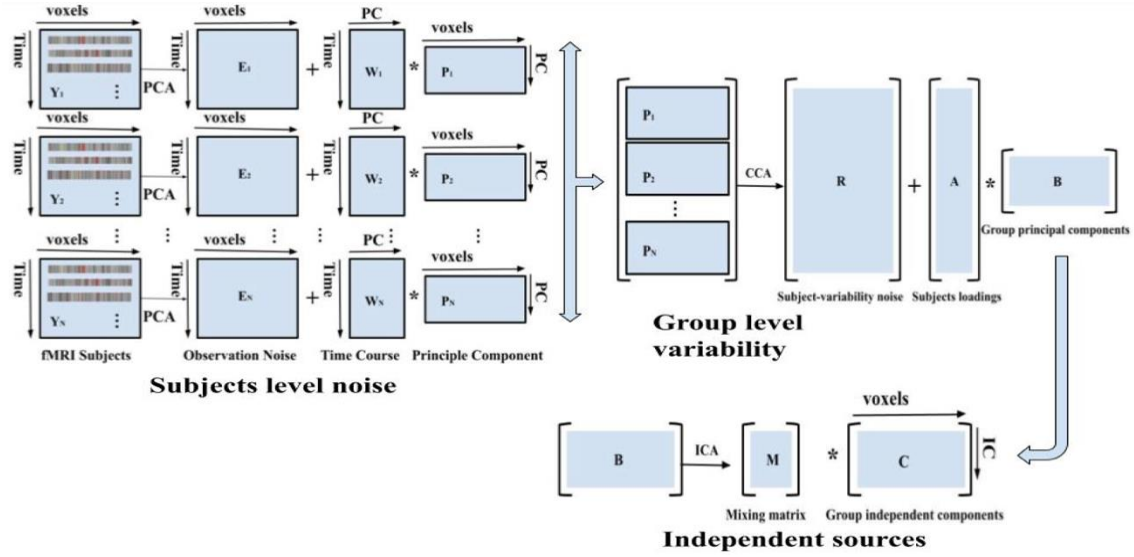


Fig 2: The steps starting from individual fMRI subjects, to obtain group-level independent components.

2.4. Dictionary Learning

Recently, dictionary learning and sparse representation have been shown to be efficient in the machine learning and pattern recognition fields [17, 18, 19, 21]. The purpose of the sparse representation is to learn a set of basis vectors and to represent the original signals using a linear combination of these basis vectors [17]. At the same time, a variety of neuroscience studies have reported the presence of sparse response in the brain neural activity [18]. The sparse response of neural activity in the brain coincides with the intrinsic nature of sparse representation methods, suggesting that sparse representation may be a possible solution to brain activity detection.

Figure 3 summarizes the framework of investigating functional networks via dictionary learning and sparse representation. Given the rs-fMRI signal matrix $\boldsymbol{S}$, where $\boldsymbol{M}$ is the total number of voxels in a subject's brain and $\boldsymbol{T}$ is the total number of fMRI time series points [19], each rs-fMRI signal in $\boldsymbol{S}$ is modelled as a linear combination of the learned basis dictionary $\boldsymbol{D}$, the model is expressed in matrix form as following:

$$\boldsymbol{S = DA}$$

, where ***A*** is holding the coefficient matrix for sparse representation. Specifically, the signal of each dictionary atom in ***D*** represents the functional activity of a specific brain network, while the vector in the coefficient matrix ***A*** represents the spatial distribution of the corresponding brain network. Finally, we identify functional network components by performing the components of interests in the learned dictionary ***D***.

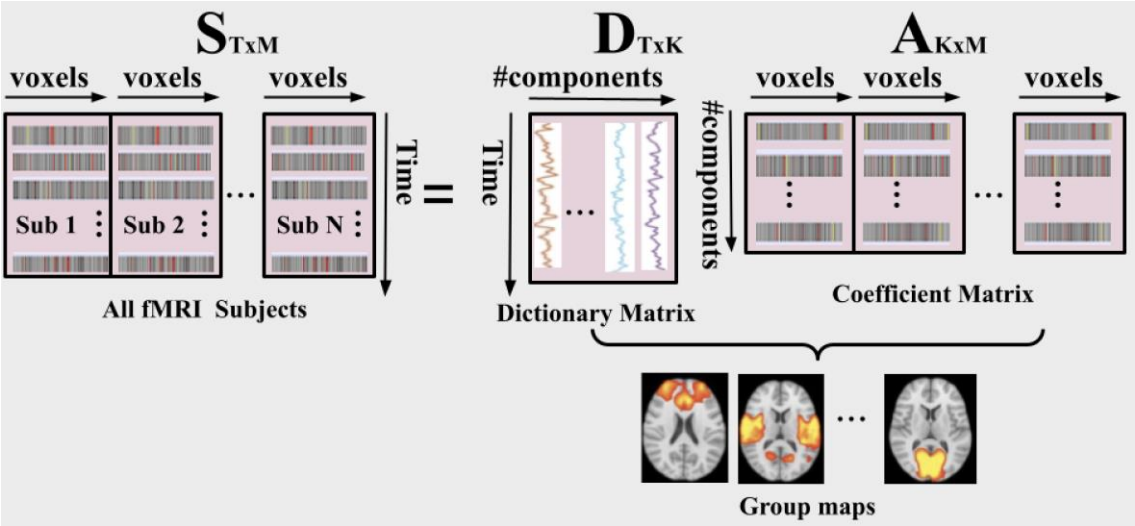


Figure 3: The framework for group-level analysis using Dictionary Learning

3. EXPERIMENTAL RESULTS

Our method and results have five main parts. First, we extracted ASD functional networks using the grouped ICA model and ranked the top 20 ROIs. Second, we extracted ASD functional networks using a dictionary learning model and ranked the top 20 ROIs. Third, we merged the 40 selected ROIs from the two models together as ASD functional networks. Fourth, we generate three corresponding masks based on the 20 selected ROIs from group ICA, the 20 ROIs selected from dictionary learning, and the 40 combined ROIs selected from both. Finally, we extract ROIs for all training sample using the above three masks, and the calculated functional connectivity was used to classify ASD and TD participants. The experimental codes for this paper are available at GitHub: https://github.com/XinYangMTSU/BIGML

3.1. Group Independent Component Analysis (group ICA)

First, the pre-processed 4D fMRI time series data from the ASD group were analyzed using the group ICA to identify spatially independent and temporally coherent networks. Figure 4 shows all 20 independent components (ROIs) derived from group ICA. Figure 5 shows each independent component individually in axial view. Figure 6 shows the ranked importance of all extracted ICA components (ROIs), and the ranking is done using the explained variance. We generate a group ICA mask based on the selected 20 ROIs.

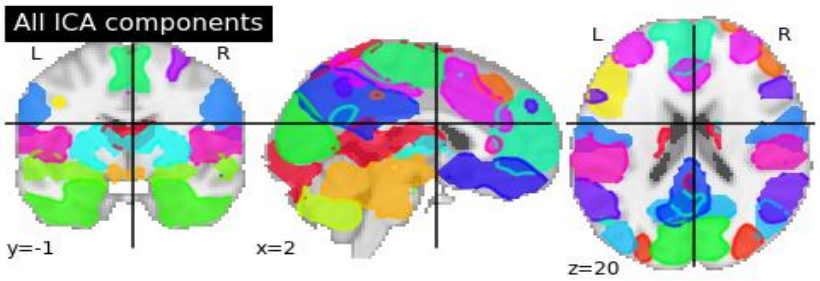


Figure 4: 20 independent components extract from group ICA

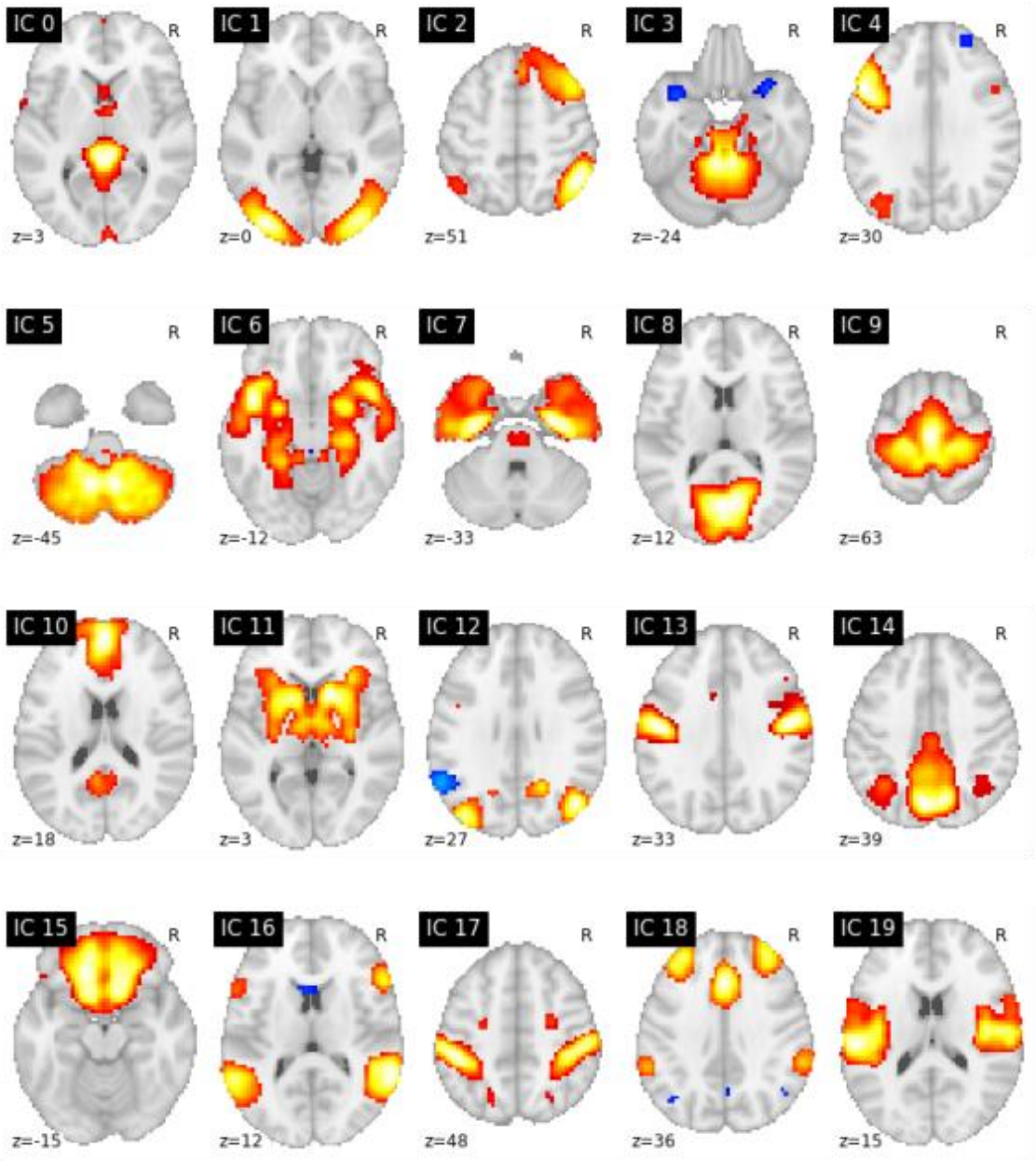


Figure 5: The 20 ICA maps derived using CanICA from rs-fMRI

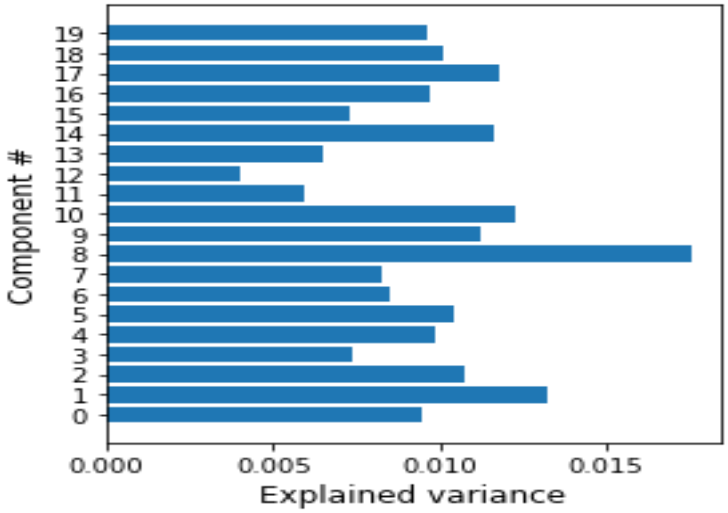


Figure 6: Explained variance of all 20 ICA components

3.2. Dictionary Learning

Second, the pre-processed 4D fMRI time series data from the ASD group were analyzed using the dictionary learning model to identify spatially independent and temporally coherent networks. Figure 7 shows all 20 independent components (ROIs) derived from dictionary learning. Figure 8 shows each independent component individually in axial view. Figure 9 shows the ranked importance of all extracted components (ROIs), and the ranking is done using the explained variance. We generate a dictionary learning mask based on the selected 20 ROIs.

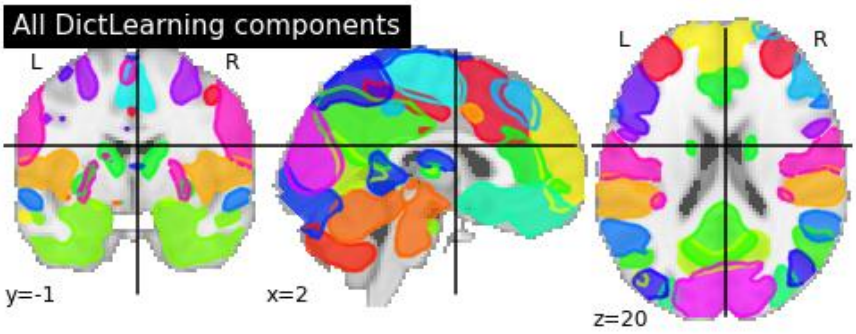


Figure 7: 20 independent components extract using dictionary learning from rs-fMRI

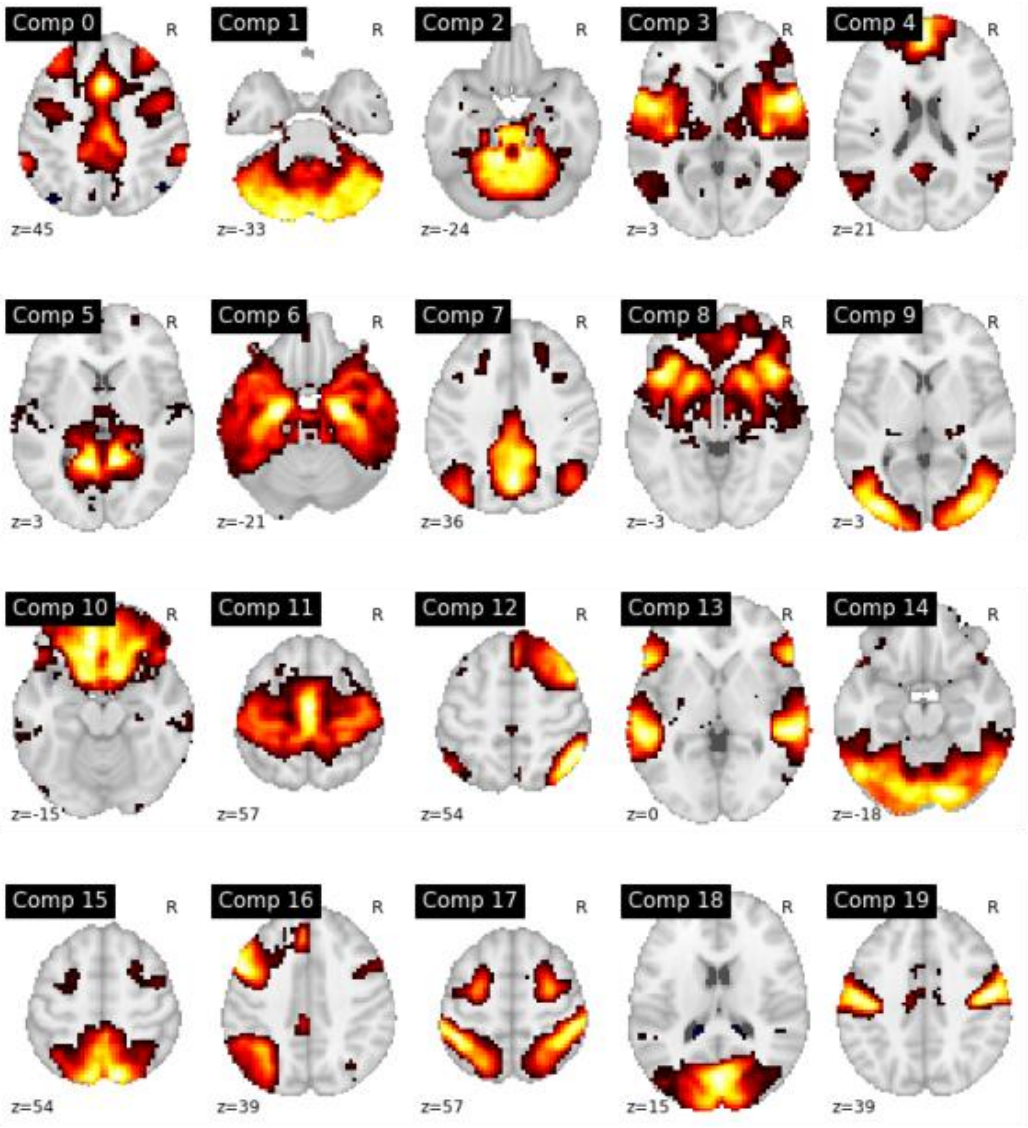


Figure 8: The 20 maps derived by dictionary learning from rs-fMRI

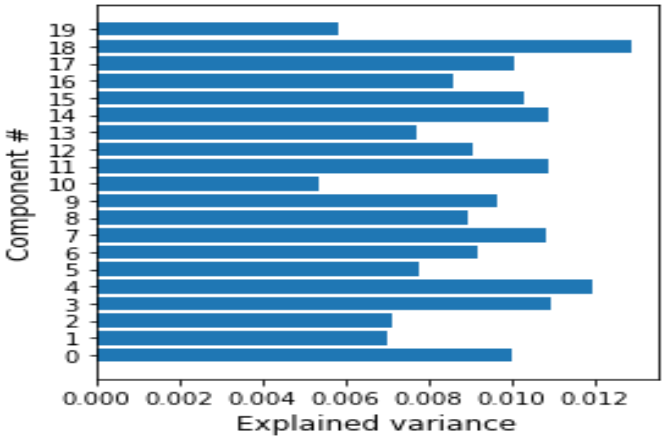


Figure 9: Explained variance of all 20 dictionary learning components

3.3. Group ICA and dictionary learning (ICA+Dict)

Third, we combined the 40 ROIs obtained from group ICA and dictionary learning together as the ASD functional networks. We generate an ICA+Dict mask based on the selected 40 ROIs.

We extracted all 40 ROIs from the rs-fMRI time series data and then calculated the functional connectivity among all ROIs. In this study, the functional connectivity calculated from the derived functional networks was used to classify ASD and TD subjects.

3.4. ASD and TD classification using functional connectivity

We generate three corresponding masks based on the 20 selected ROIs from group ICA, the 20 ROIs selected from dictionary learning, and the 40 combined ROIs selected from both. Based on the above three generated masks, we can extract corresponding ROIs for all training samples and calculate the pairwise functional connectivity. The calculated functional connectivity was used to classify ASD and TD participants.

In order to calculate the connectivity matrix of the ROIs, we can implement the Pearson correlation. Every functional connectivity feature in the connectivity matrix is represented by a Pearson correlation coefficient, which is used to measure the mutual relationship between two brain regions of interest. The threshold of Pearson's correlation coefficient ranges from -1 to +1. When the threshold value of the Pearson correlation coefficient approaches -1, this indicates an opposite association between the two brain regions; in contrast, if the Pearson coefficient value approaches 1, it indicates a high correlation between the two brain regions. It is worth noting that the Pearson correlation connectivity matrix is symmetric, which means that the corresponding upper and lower triangular values are the same. Therefore, we only need to use the upper triangular or lower triangular values in the correlation matrix as features for the ASD and TD classification. In addition to this, the main diagonal in the connectivity matrix should also be removed, as these values represent self-correlated regions.

For single group ICA and single dictionary learning, the ROI-based functional connectivity of the extracted 20 functional ROIs was calculated, resulting in 190 ($\frac{(400-20)}{2} = 190$) pair-wise connectivity features for each subject. This set of 190 features for each subject was used as features for ASD and TD classification.

For ICA+Dict, the ROI-based functional connectivity of the extracted 40 functional ROIs was calculated, resulting in 780 ($\frac{(1600-40)}{2} = 780$) pair-wise connectivity features for each subject. This set of 780 features for each subject was used as features for ASD and TD classification.

In our experiments, we use the Gaussian kernel support vector machine (kSVM) to classify ASD and TD. To evaluate the performance, we did five-fold cross-validation. Table 2 shows that the 5-fold cross-validation sensitivity, specificity and accuracy of ICA+Dict are better than that of single ICA or single dictionary learning. It is noteworthy that the sensitivity of ICA+Dict is improved by at least 3% compared to single ICA or single dictionary learning. For ASD studies, sensitivity is a more important metric than specificity. Therefore, overall, ICA+Dict outperforms either single ICA or single dictionary learning. Figure 10, 11 and 12 shows the five-fold cross-validation receiver operating characteristic (ROC) curve for group ICA, dictionary learning, and ICA+Dict respectively.

Table 2: Five-fold Cross-Validation Results

	Sensitivity	Specificity	Accuracy
kSVM-ICA	55.61%	58.08%	56.67%
kSVM-Dict	65.76%	69.10%	67.50%
kSVM-ICA+Dict	**69.24%**	**69.23%**	**69.17%**

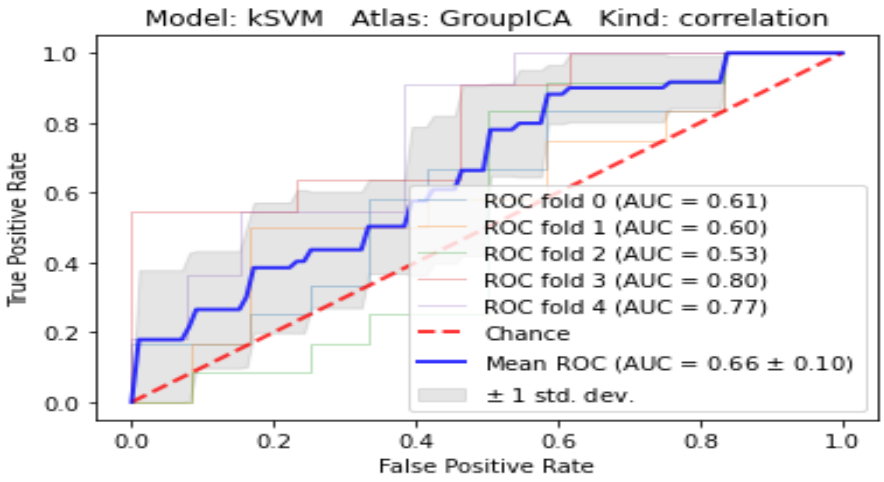


Figure 10: Five-fold cross validation ROC Curve for groupICA

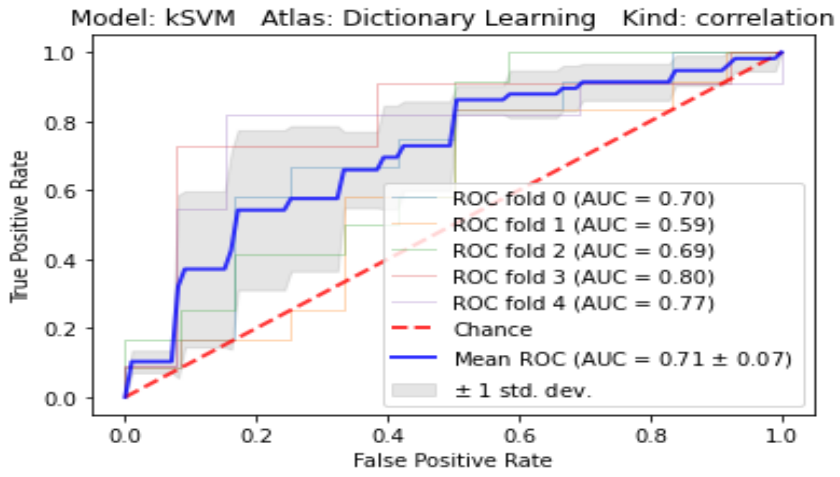


Figure 11: Five-fold cross validation ROC Curve for Dictionary Learning

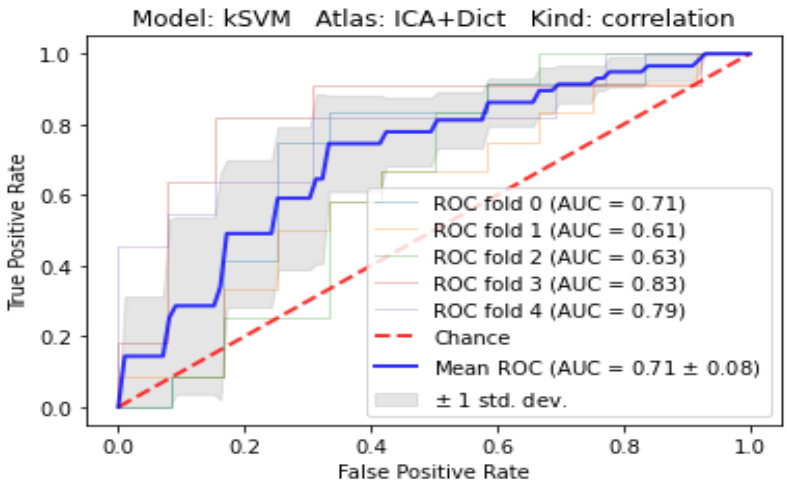


Figure 12: Five-fold cross validation ROC Curve fro group ICA and Dictionary Learning

4. CONCLUSIONS

In this study, we used group ICA and dictionary learning model together to derive functional networks for the autism spectrum disorder (ASD) population, and the functional connectivity calculated from the derived functional networks is used as the feature to classify ASD and TD participants. We combined the 40 ROIs obtained from group ICA and dictionary learning together as the ASD functional networks. The ROI-based functional connectivity of the extracted 40 functional ROIs was calculated, resulting in 780 pair-wise connectivity features for each subject. This set of 780 features for each subject was used as features for ASD and TD classification. The 5-fold cross-validation results showed that the functional networks derived from ICA and dictionary learning together obtain higher sensitivity, specificity and accuracy than a single ICA model or single dictionary learning model. Our results demonstrate that the improved data-driven algorithm can extract ROIs containing important information to improve the accuracy of ASD and TD classification. This also shows that data-driven algorithm is still a promising research direction for the field of ASD research.

In this experiment, our sample size of 120 is still relatively small. To get more accurate and stable results, we need to collect more samples to conduct the experiment. Nowadays, deep learning has been widely used in various fields. Nonetheless, supervised deep learning requires a training sample set of sufficient size. Our further research in ASD classification will require the need to collect a large amount of training data to achieve considerable breakthroughs in sensitivity, accuracy, and specificity.

REFERENCES

[1] Poldrack, R.A., Mumford, J.A. and Nichols, T.E., 2011. Handbook of functional MRI data analysis. Cambridge University Press.

[2] Poldrack RA, Mumford JA, Nichols TE. Handbook of functional MRI data analysis. Cambridge University Press; 2011 Aug 22.

[3] Biswal, B., ZerrinYetkin, F., Haughton, V.M. and Hyde, J.S., 1995. Functional connectivity in the motor cortex of resting human brain using echo-planar MRI. Magnetic resonance in medicine, 34(4), pp.537-541.

[4] Biswal BB, Van Kylen J, Hyde JS. Simultaneous assessment of flow and BOLD signals in resting-state functional connectivity maps. NMR in Biomedicine. 1997 Jun-Aug;10(4-5):165-170.

[5] Damoiseaux, Jessica &Rombouts, Serge &Barkhof, Frederik &Scheltens, Ph & Stam, C.J. & Smith, S.M. & Beckmann, Christian. (2006). Consistent resting-state networks. Proceedings of the National Academy of Sciences of the United States of America. 103. 13848-53. 10.1073/pnas.0601417103.

[6] Greicius MD, Srivastava G, Reiss AL, Menon V. Default-mode network activity distinguishes Alzheimer's disease from healthy aging: evidence from functional MRI. Proc Natl Acad Sci U S A. 2004 Mar 30;101(13):4637-42. doi: 10.1073/pnas.0308627101. Epub 2004 Mar 15. PMID: 15070770; PMCID: PMC384799.

[7] Al-Zubaidi, A., Mertins, A., Heldmann, M., Jauch-Chara, K. and Münte, T.F., 2019. Machine learning based classification of resting-state fMRI features exemplified by metabolic state (hunger/satiety). Frontiers in human neuroscience, 13, p.164.

[8] McClure I. Volkmar FR, Paul R, Rogers SJ, Pelphrey KA, editors. Handbook of autism and pervasive developmental disorders, fourth edition. Hoboken, NJ: John Wiley & Sons; 2014

[9] Just, Marcel Adam, Vladimir L. Cherkassky, Timothy A. Keller, and Nancy J. Minshew. "Cortical activation and synchronization during sentence comprehension in high-functioning autism: evidence of underconnectivity." Brain 127, no. 8 (2004): 1811-1821.

[10] Courchesne, Eric, Karen Pierce, Cynthia M. Schumann, Elizabeth Redcay, Joseph A. Buckwalter, Daniel P. Kennedy, and John Morgan. "Mapping early brain development in autism." Neuron 56, no. 2 (2007): 399-413.

[11] Kleinhans, Natalia M., Todd Richards, Lindsey Sterling, Keith C. Stegbauer, Roderick Mahurin, L. Clark Johnson, Jessica Greenson, Geraldine Dawson, and Elizabeth Aylward. "Abnormal functional connectivity in autism spectrum disorders during face processing." Brain 131, no. 4 (2008): 1000-1012.

[12] Weng, Shih-Jen, Jillian Lee Wiggins, Scott J. Peltier, Melisa Carrasco, Susan Risi, Catherine Lord, and Christopher S. Monk. "Alterations of resting-state functional connectivity in the default network in adolescents with autism spectrum disorders." Brain research 1313 (2010): 202-214.

[13] Comon, P., 1994. Independent component analysis, a new concept?. Signal processing, 36(3), pp.287-314.

[14] Tošić, I. and Frossard, P., 2011. Dictionary learning. IEEE Signal Processing Magazine, 28(2), pp.27-38.

[15] Stone, J.V., 2004. Independent component analysis: a tutorial introduction.

[16] Bordier, C., Dojat, M. and de Micheaux, P.L., 2011. Temporal and spatial independent component analysis for fMRI data sets embedded in the AnalyzeFMRI R package. Journal of Statistical Software, 44(9), pp.1-24.

[17] Zhao, S., Han, J., Lv, J., Jiang, X., Hu, X., Zhao, Y., Ge, B., Guo, L. and Liu, T., 2015. Supervised dictionary learning for inferring concurrent brain networks. IEEE transactions on medical imaging, 34(10), pp.2036-2045.

[18] Lv, J., Jiang, X., Li, X., Zhu, D., Chen, H., Zhang, T., Zhang, S., Hu, X., Han, J., Huang, H. and Zhang, J., 2015. Sparse representation of whole-brain fMRI signals for identification of functional networks. Medical image analysis, 20(1), pp.112-134.

[19] Mairal, J., Bach, F., Ponce, J. and Sapiro, G., 2010. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research, 11(1).

[20] Varoquaux, G., Sadaghiani, S., Pinel, P., Kleinschmidt, A., Poline, J.B. and Thirion, B., 2010. A group model for stable multi-subject ICA on fMRI datasets. Neuroimage, 51(1), pp.288-299.

[21] Mensch, A., Varoquaux, G. and Thirion, B., 2016, April. Compressed online dictionary learning for fast resting-state fMRI decomposition. In 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI) (pp. 1282-1285). IEEE.

[22] Calhoun, V.D., Adali, T., Pearlson, G.D. and Pekar, J.J., 2001. A method for making group inferences from functional MRI data using independent component analysis. Human brain mapping, 14(3), pp.140-151.

[23] Beckmann, C.F., DeLuca, M., Devlin, J.T. and Smith, S.M., 2005. Investigations into resting-state connectivity using independent component analysis. Philosophical Transactions of the Royal Society B: Biological Sciences, 360(1457), pp.1001-1013.

EFFECTS OF NONLINEAR FUNCTIONS ON KNOWLEDGE GRAPH CONVOLUTIONAL NETWORKS FOR RECOMMENDER SYSTEMS WITH YELP KNOWLEDGE GRAPH

Xing Wei and Jiangjiang Liu

Department of Computer Science, Lamar University, Beaumont, USA

ABSTRACT

Knowledge Graph (KG) related recommendation method is advanced in dealing with cold start problems and sparse data. Knowledge Graph Convolutional Network (KGCN) is an end-to-end framework that has been proved to have the ability to capture latent item-entity features by mining their associated attributes on the KG. In KGCN, aggregator plays a key role for extracting information from the high-order structure. In this work, we proposed Knowledge Graph Processor (KGP) for pre-processing data and building corresponding knowledge graphs. A knowledge graph for the Yelp Open dataset was constructed with KGP. In addition, we investigated the impacts of various aggregators with three nonlinear functions on KGCN with Yelp Open dataset KG.

KEYWORDS

Recommender Systems, Knowledge Graph, Activation Function.

1. INTRODUCTION

In 2019, the number of internet users reached 7.71 billion [1], and 2.5 quintillion bytes of data was being created every day [2]. It becomes more and more challenging for people and companies to cope with such dramatic data explosion. For example, Netflix, the online streaming-service provider, offers more than 10 thousand movies and TV shows from which users can choose. The traditional search method only displays the sorted list of items relating to the search key word and cannot provide specific items in different users' interests, so the users may not find the items they really want. The heavy information overload has become a major problem for both consumers and providers of the online content industry.

To better deliver content to users, one of the practical strategies is personalization. As a successful solution, the recommender systems have been playing a vital and indispensable role in Web applications, ranging from search engines and E-commerce to social media sites and online streaming services. Almost every online content provider has applied a recommender system.

The traditional recommendation methods, such as Collaborative Filtering (CF) and content-based recommendation, have achieved good performance on rating data. However, despite CF's effectiveness and universality, its ability on modeling side information, such as item attributes and user profiles [3], suffers from the cold start problem, which happens when items added to the catalogue have either none or very little interactions and consequently process sparse data where

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 185-199, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110715

users and items have few interactions. As a common solution for those problems, model-based methods are designed to transform user ID, item ID, and the side information into a generic feature vector that compensates for the sparse data and improves the recommendation performance, such as matrix factorization [4], factorization machine (FM) [5], and Wide & Deep [6].

However, these methods only view each interaction between entities as an independent data instance rather than linked data with relations. This makes them insufficient to distill attribute-based collaborative signals from the collective behaviors of users. As we can see in Figure 1, there is an interaction between User John and Movie m2, which is directed by Director D1, and Director D1 directed Movie m2 and Movie m4. CF methods can only determine the similarity of users who also watched Movie m2, such as User David and User Paul. Model-based methods find the similar items Movie m21 and Movie m2 by the same attributes of Actor A1 and Director D1.As we can see, based on these two types of information, not only recommendation can be generated, but also a high-order relationship can be found, which connects user and item with one or multiple linked attributes.

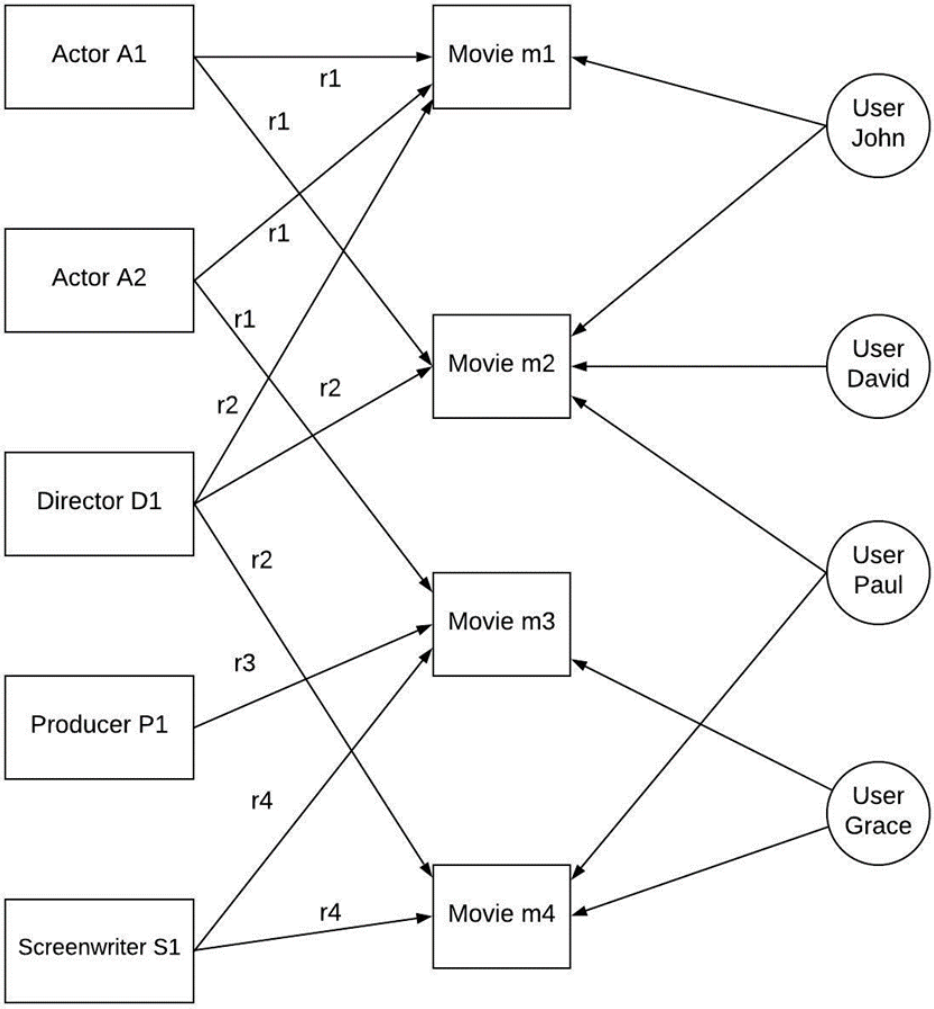


Figure 1. A movie example of knowledge graph. Relations r1: actor of, r2: director of, r3: producer of, r4: screenwriter of.

Therefore, to fulfill the shortage of traditional recommendation methods, graphsare a solution to collaborating with side information. Knowledge graphs (KGs) are composed of structured information of the real world, which links attributes themselves as well as with users and items. Typically, a knowledge graph is a directed heterogeneous graph in which each edge is represented as a triple (head entity, relation, tail entity), indicating that two entities are connected by a specific relation, e.g., (George Lucas, film, director, Star Wars). In general, because we can process the information from the high-order structure of the knowledge graph, by comparing with the traditional model method, the knowledge graph related method is advanced in dealing with cold start problems and sparse data.

Several recent efforts have applied knowledge graphs on the recommendation system. For example, knowledge graph embedding methods, which transfer entities and relations to low-dimensional representation vectors [21], and graph algorithm-based methods, which exploit the latent information from users, items, and the relations in between them by treating knowledge graph as a high-order structure information network [22]. KGCN is an end-to-end framework that captures latent item-entity features by mining their associated attributes from the high-order relationships on the KG. Additionally, an aggregator plays an important role in the KGCN. In the field of graph-related neural networks, aggregators widely operate with nonlinear functions such as ReLU [7], Leaky ReLU [8], and Tanh [9].

1.1. Our Contributions

To build a knowledge graph from scratch often requires tremendous time and effort, which is an obstacle for most researchers studying knowledge graphs. The process of adopting knowledge graphs for business analytics to support decision making is even more challenging and time consuming.

To tackle this challenge, we propose an efficient tool, Knowledge Graph Processor (KGP), with a user-friendly interface that can easily transfer the raw dataset to a knowledge graph format dataset. We built a Yelp knowledge graph from a Yelp Dataset by using KGP.

We also conducted analysis on aggregators used in the Knowledge Graph Convolutional Network, with several widely adopted nonlinear functions and achieved significant improvement, compared to the original KGCN with ReLU aggregators.

We released the code of KGP and Yelp Knowledge Graph datasets (knowledge graphs). The source code and the dataset are available at https://github.com/XingWeiLamar/KGP.

2. RELATED WORK

2.1. Knowledge Graph

A knowledge graph is a multi-relational graph constructed by entities (nodes) and relations (different types of edges). Each edge represents a triple of the form (head entity, relation, tail entity). Recently, the knowledge graph (KG) has been rapidly applied to various applications.

Large-scale knowledge graphs for academic and commercial purposes, such as NELL, DBpedia, Google Knowledge Graph, and Microsoft Satori, have become the core data structure of many practical applications from named entity disambiguation [20] and information extraction [17] to search engines [18] and question/answer systems [10].

Traditional recommendation techniques, such as collaborative filtering, usually represent customer-items interaction as an N-dimensional vector, then model their interaction by specific techniques, such as inner product or neural networks. However, CF methods usually suffer from limited performance when user-item interactions are very sparse, and they perform poorly when processing new products and users. To address those limitations, a common paradigm is to turn the user-item interaction into a more feature-based scenario, where attributes of users and items are transformed into the model as vectors to remedy the sparsity and perform better in cold start scenarios. [6]

2.2. Recent Knowledge Graph Related Recommender System

The successful applications of knowledge graphs in a wide variety of tasks have inspired researchers to study KG on improving the performance of recommendation systems. In comparison with knowledge-free methods, applying knowledge graphs on recommender system gains advantages in three ways: 1) discovering latent connections among items of knowledge graphs by the semantic relatedness among items; 2) exploring users' interests to increase the diversity of the recommendation from the varied types of relations; and 3) improving the exam inability of recommender systems due to the connections of user's historically-liked and recommended items in the knowledge graph.

There are three types of knowledge-aware recommender systems, based on current research. First, embedding-based methods [19] pre-process a knowledge graph with knowledge graph embedding algorithms then process user entity embedding into recommendation. Embedding-based methods can easily utilize KGs to fulfill the needs of recommender systems. However, the knowledge graph embedding algorithms are designed to model rigorous semantic relatedness [13], which can perform much better on graph applications (e.g., link prediction) rather than recommendation systems. Furthermore, embedding-based methods do not usually do end-to-end way training. Second, path-based methods [11] provide instruction for recommendation from discovered patterns of connections among entities in a KG. It is barely applied in practical terms from a cost and effectiveness perspective because meta-paths/meta-graphs need to be manually designed. Third, hybrid methods [12] combine the former two methods and learn user-item embeddings by extracting the structure of knowledge graphs.

2.3. Nonlinear Functions

For Neural Networks, the purpose of the nonlinear activation function is to introduce non-linearity into the output. A neural network without a nonlinear activation function is essentially a linear regression model. The activation function does the non-linear transformation to the input-making so it is capable of learning and performing more complex tasks.

The most common nonlinear activation function is ReLU, which is defined as: $\sigma(x) = \max\{0, x\}$. Hence, whenever x is negative, the function returns 0. When X is positive, it returns x. Note that this function is not differentiable at 0, and hence the back propagation will fail at this point. Nonetheless, in general, because of the precision issues of floating-point numbers in computers, this situation barely appear in reality, and ReLU as defined above works well in practice.

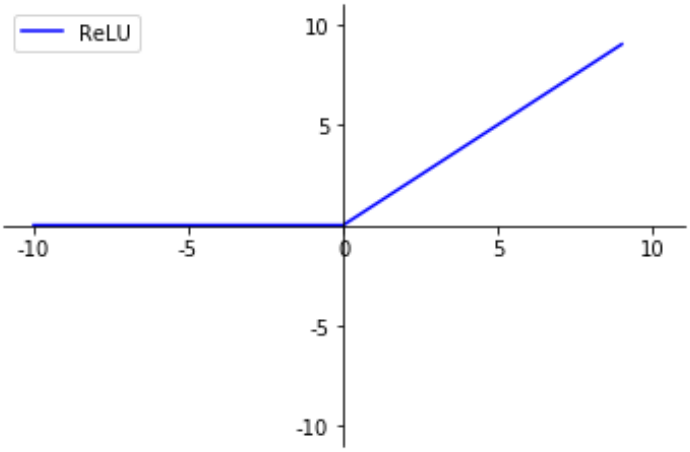


Figure 2. ReLU

One of the variations of ReLU is Leaky-ReLU [1], which is given by: $\sigma(x) = x$ if $x > 0$ and $a \cdot x$ otherwise, in which a is an adjustable constant. Instead of defining the ReLU function as 0 for negative values of x, it is defined as an extremely small linear component of x. By making this small modification, the gradient of the left side of the graph comes out to be a non-zero value. Hence, we would no longer encounter dead neurons in that region.

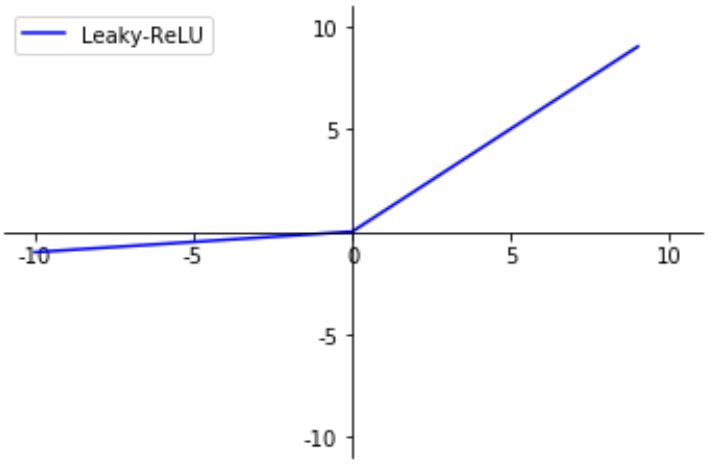


Figure 3. Leaky-ReLU

Exponential Linear function is also a special case of ReLU. The function is given by: $\sigma(x) = x$ for $x > 0$ and $a \cdot (e^{x} - 1)$ for $x<0$, where a is a hyper-parameter to be learned from the data. Unlike the Leaky-ReLU and parametric ReLU functions, instead of a straight line, ELU uses a log curve for defining the negative values.

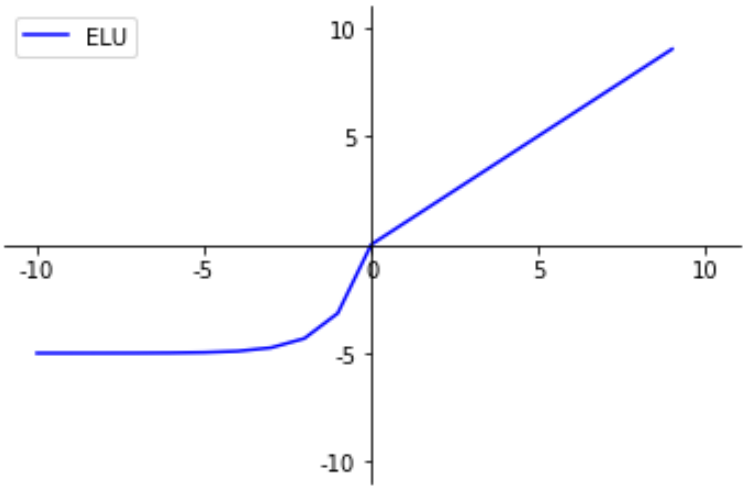


Figure 4. ELU

3. PROPOSED WORK

3.1. Knowledge Graph Processor

Data analysts often spend most of their time on data pre-processing. Data scientists mostly spend 80 percent of their time on finding, cleansing, and organizing data. [16] In the experiment, we utilized the Yelp dataset as a benchmark, which is publicly accessible and varies in terms of domain, size, and sparsity.

Therefore, to conquer this problem, we developed an easy-to-use tool, Knowledge Graph Processor, KGP, which is able to pre-process the raw dataset efficiently and effectively. The workflow chart of KGP is shown in Figure 5. First, KGP detects the format of the dataset. If it is stored in JSON format, KGP will transfer the JSON file to CSV format. Next, KGP performs a data cleaning function to remove the duplicated values and null value in the dataset. Then, users will input the relation's name and assign the head and tail of the knowledge graph triplet by selecting the column number in the dataset, so KGP can extract the selected data from those columns and construct them with relation named as a triplet. After users finish the triplets building, KGP extracts those triplets and reconstructs them into a CSV file and assigns an index value for each item. Finally, after users input a user-item interaction file, which usually is the file stores rating information, the KGP transforms the rating file from explicit feedback to implicit feedback. (explanation is in section 3.2)

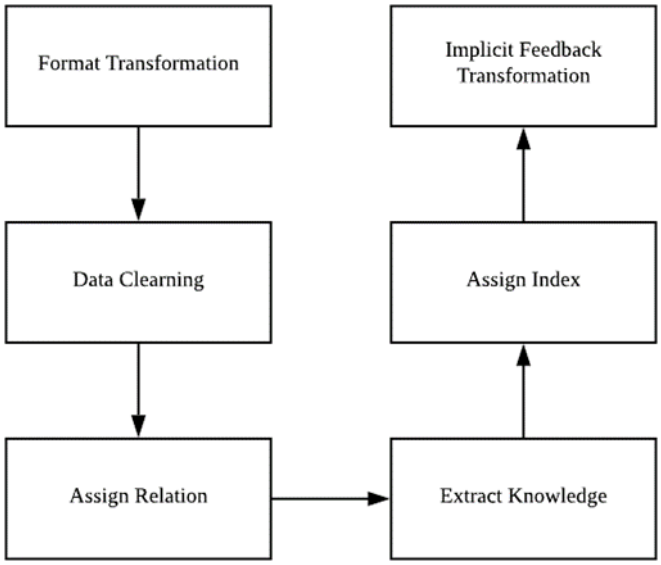


Figure 5. Working Flow of KGP

3.2. Yelp Knowledge Graph with KGP

By using KGP, we constructed a Yelp knowledge graph from Yelp open dataset[1] which contains over 6 million reviews, nearly 20 thousand businesses. The local businesses such as restaurants, bars, spas and barber shops are viewed as items.

Beside the item-to-item interactions, we extracted item knowledge from the local business information network (e.g., categories, locations, and attributes) as knowledge graph data. To ensure the knowledge graph quality, we pre-processed the three knowledge graph parts by filtering out entities with frequency lower than 10, and retaining the relations appearing in at least 50 triplets. The basic statistics of the datasets are presented in Table 1.

To reduce the noise deviation and test the performance of the model in a different sparsity, we use 10-core, 20-core, and 30-core settings to ensure that each item has at least 10 interactions, 20 interactions, and 30 interactions in the knowledge graph.

The original rating dataset is explicit feedback, in which the rated stars are from 0-5, with 5 is the best and 0 as the worst. From the rating stars, we can directly understand how much the user likes or dislikes the business. However, the practical data behaviors are implicit feedback, which are more complicated and important. We transform the explicit feedback into implicit feedback where the entry will be marked with 1, indicating that the user rated the item positively, with the positive rating threshold as 4, and sample an unwatched set marked as 0 for each user.

3.3. Problem Formulation

In this study, there is a set of *M*of users $U = \{u_1, u_2..., u_m\}$ and a set of *N* of items $I = \{i_1, i_2, ...,i_n\}$. The user-item interaction matrix $P \in R^{m \times n}$ defined as users' implicit feedback, which $y_{ui}=1$indicates that user *u* engages with item *v*, such as, browsing and clicking, if not $y_{ui}=0$. Moreover, the knowledge graph G is defined as entity-relation-entity triples *(h,r,t)*. Here $h \in E$,

[1]https://www.yelp.com/dataset

$r \in R$, and t∈E infer the head, relation, and tail. *E* and *R* are the set of entities and relations in the knowledge graph. For example, the triple (Star Wars: The Rise of Skywalker, movie; director; J.J. Abrams) indicates the fact that J.J. Abrams is the director of the movie *Star Wars: The Rise of Skywalker*. In our scenarios, an item $i \in I$ correspond to one entity $e \in E$. For instance, in movie recommendations, the knowledge also contains the item "Star Wars: The Rise of Skywalker" as an entity.

The KGCN model is to predict whether user *u* has potential interest in new item *i* by given the user-item matrix *P* and the knowledge graph *G*. We have a prediction function $f_{ui}=F(u,v|\Theta,Y,G)$, where P_{ui} denotes the probability that user *u* will engage with item *i*, and Θ denotes the model parameters of function *F*.

3.4. Analysis of Three Aggregators for Knowledge Graph Convolutional Networks

Knowledge Graph Convolutional Networks [14] is an end-to-end framework that explores users' preferences on knowledge graph for recommender systems. The Architecture KGCN is shown as Figure 6, where the first KGCN layer captures the high-order structural proximity by aggregating each entity's representation and its neighborhood representation into a single vector. Then the Learning Layer will take the H-order entity representation fed into a function $R^d x R^d \rightarrow R$ to predict probability.

First, consider a candidate pair of user *u* and item(entity) *v*. *N(v)* is denoted as the set of entities directly connected to *v*, and $r_{ei,ej}$ is denoted as the relation between entity e_i and e_j. The function: $R^d x R^d \rightarrow R$ is used to calculate the score between a user and a relation:

$$\pi_v^u = g(u,r) \qquad (1)$$

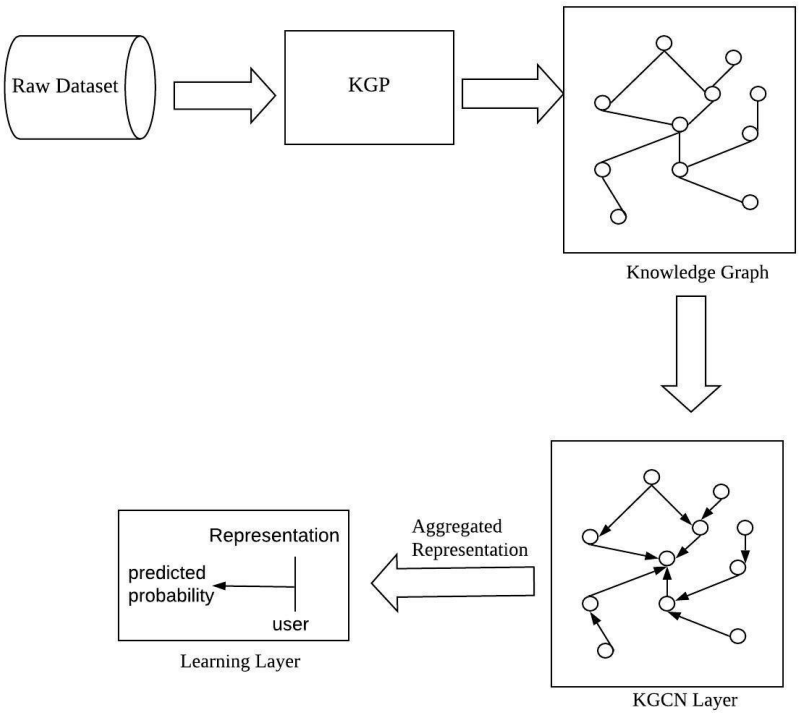


Figure 6. KGP+KGCN Architecture

In the formula, $u \in R^d$ and $r \in R^d$ are the representations of user u and relation r, and d is the dimension of representations. The function is to compute the importance between relation r and user u. For instance, a user might have more interests in restaurants in specific categories and another user may be more concerned about whether or not the restaurant is kid-friendly.

The below linear function combines v's neighborhood to capture the topological proximity structure of item v.

$$V^u_{N(v)} = \sum_{e \in N(v)} \tilde{\pi}^u_{r_{v,e}} e' \qquad (2)$$

$\tilde{\pi}^u_{r_{v,e}}$ is the normalized user-relation score:

$$\tilde{\pi}^{u}_{r_{v,e}} = \frac{\exp(\pi^{u}_{r_{v,e}})}{\sum_{e \in N(v)} \exp(\pi^{u}_{r_{v,c}})} \qquad (3)$$

The e denotes the representation of entity e. User-relation scores perform as personalized filters in the formula 2 because we aggregate the neighbors with bias, with respect to these user-relation scores.

Next, KGCN layer aggregates the entity representation v and its neighborhood representation $V^u_{S(v)}$ into a single vector by aggregators. ReLU is the default aggregator used in KGCN.

Aggregators perform a significant role in the KGCN, and different non-linear functions can affect the output of the KGCN. In this study, we analyze the effectiveness of three aggregators with the following different nonlinear function.

$$\mathrm{agg}_{\mathrm{Sum}} = \sigma(w \cdot (v + v^{u}_{s(v)}) + b) \qquad (4)$$

$$\mathrm{agg}_{\mathrm{concat}} = \sigma(w \cdot \mathrm{concat}(v, v^{u}_{s(v)}) + b) \qquad (5)$$

$$\mathrm{agg}_{\mathrm{neighbor}} = \sigma(w \cdot v^{u}_{s(v)} + b) \qquad (6)$$

where σ denotes non-linear functions, w and b are transformation weight and bias.

The KGCN layer extends 1-order entity representation to h-order entity representation by iterating the KGCN layer multiple times, so the algorithm can explore the user's interests more comprehensively. To achieve that, the KGCN layer first propagates the initial representation of each entity to its neighbors and aggregates the vectors to receive the 1-order representations. Then it repeats the procedure and aggregates the 1-order representation and its neighbors to obtain the 2-order representations. The combination of an entity and its neighbors up to h hops away is the h-order representation of an entity.

The KGCN learning algorithm is as the following. The algorithm receives a pair of users and items then calculates the receptive field M of v layer by layer. Next the aggregation repeats H times: in iteration h, it calculates the neighborhood representation of each entity $e \in M[h]$ then aggregates it with its own representations $e^u[h-1]$ to achieve the representation to be used in the next iteration. Last, the final H-order entity representation vu will be pass to the prediction function f: $R^d \times R^d \rightarrow R$ with user representation u.

$$\hat{y}_{uv} = f(u, v^u) \quad (7)$$

4. EXPERIMENTS

4.1. Datasets

Yelp open dataset contains over 6 million reviews, nearly 20 thousand businesses.

MovieLens-20M is a widely used benchmark dataset for movie recommendation, which contains over 20 million explicit ratings (rating from 1-5). The knowledge graph of MovieLens-20M is pre-constructed by Wang with Microsoft Satori [14].

Table 1. Statistics of the datasets

	10-Core	20-Core	30-Core	MovieLens-20M
Number of Users	44837	44784	16295	138159
Number of Items	71822	26613	2550	16954
Number of interactions	876829	320872	24611	13501622
Number of Entities	70534	25509	1822	102569
Number of Relations	34	34	33	32
Number of KG Triples	1268941	627552	60463	499474

4.2. Data Pre-processing

We built a Yelp Knowledge Graph from a Yelp open dataset. To reduce the noise deviation and test the performance of the model in different sparsity, we use 10-core, 20-core, and 30-core settings to ensure that each item has at least 10 interactions, 20 interactions, and 30 interactions in the knowledge graph. Please see the statistics of each datasets in Table 1.

4.3. Nonlinear Function Setting

In the experiment, to achieve the best performance, we set a as 0.2 for the Leaky ReLU. For the ELU a is 1.

4.4. Baseline

We tested the dataset with 3 baselines, in which the first baseline is a well-known KG-free baseline, the second one is a KG-aware method, and the third one is the KGCN with ReLU function.

LibFM [7] is a software implementation for factorization machines that features stochastic gradient descent (SGD) and alternating least squares (ALS) optimization as well as Bayesian inference using Markov Chain Monte Carlo (MCMC).

RippleNet [15] is a memory-network-like approach propagating user preferences over the entities in KG and iteratively extending a user's potential interest along links in the KG to provide recommendation.

4.5. Experiments Setup

In KGCN, the function g and f are set as inner product, σ is non-linear function for non-last-layer aggregator, and tanh for last-layer aggregator. We applied hyper-parameters as follows:

Each hyper-parameter is determined by the best Accuracy (AUC) on the corresponding dataset. For each dataset, we randomly select 80% of rating interaction history as the training set and 20% as the evaluation set, as well as a test set, respectively. Each experiment is repeated three times, and the average outcome is reported. The performance of each method is evaluated by two evaluation criteria: AUC and F1, which are applied to evaluate the click through rate (CTR) prediction. The test set was applied in training model to predict each interaction.

For the baseline, we use the Python vision of LibFm, Pylibfm to test the dataset. The number of factors is 10, the number of training epochs is 50, and the initial learning rate is 1×10-4. For the RippleNet, dimension of embeddings is 8, number of hops is 2, learning rate is 0.02, l2 weight is 1×10-7, batch size is 1024. For the KGCN ReLU λ is 3×10-8, μ is 0.003, batch size is 1024, H is 8, d is 64, S is 8.

Table 2. Basic Hyper Parameter Setting for KGCN

	10-Core	**20-Core**	**30-Core**	**MovieLens-20M**
Dimension of embeddings(d)	**128**	**64**	**128**	**32**
Neighbor Sampling size(S)	**8**	**8**	**8**	**4**
Depth of receptive field(H)	**2**	**2**	**2**	**2**
L2 Regularize Weight(λ)	2×10^{-9}	2×10^{-9}	3×10^{-5}	2×10^{-7}
Learning Rate(μ)	3×10^{-3}	3×10^{-3}	6×10^{-4}	2×10^{-2}
Batch Size	**1024**	**1024**	**1024**	**65536**

5. RESULT

The results of click through rate are presented in Table 3. We have the following observations:

Among all the models, the KGCN Leaky ReLU-sum achieves the best performance on the average of the three datasets.

In general, comparing with KG free model LibFM, we find that the improvements of KGCN Leaky ReLU-sum on 10-core and 20-core are 15% and 13% higher than 30-core. This demonstrates that KGCN Leaky ReLU-sum can well addres0.80.s sparse scenarios, since the 10-core and 20-core datasets are sparser than 30-core.

RippleNet shows strong performance compared with the KG-free baseline, because RippleNet also uses a multi-hop neighborhood structure, which also captures the proximity information from the KG.

Table 3. The results of AUC and F1 in CTR prediction

	10-Core		20-Core		30-Core		MovieLens-20M	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
PyLibFm	0.715	0.674	0.723	0.671	0.702	0.655	0.955	0.903
RippleNet	0.903	0.863	0.897	0.855	0.721	0.639	0.968	0.912
KGCN-Leak-ReLU-Sum	0.914	0.854	0.896	0.835	0.791	0.674	0.979	0.934
KGCN-Leak-ReLU-Concat	0.913	0.850	0.903	0.841	0.768	0.662	0.978	0.933
KGCN-Leak-ReLU-Neighbor	0.838	0.774	0.814	0.753	0.484	0.595	0.977	0.932
KGCN-ELU-Sum	0.906	0.842	0.895	0.833	0.754	0.629	0.977	0.931
KGCN-ELU-Concat	0.902	0.840	0.903	0.841	0.708	0.725	0.978	0.932
KGCN-ELU-Neighbor	0.835	0.755	0.814	0.753	0.665	0.582	0.976	0.932
KGCN-ReLU-Sum	0.912	0.851	0.891	0.833	0.765	0.650	0.978	0.932
KGCN-ReLU-Concat	0.912	0.852	0.905	0.842	0.773	0.666	0.977	0.931
KGCN-ReLU-Neighbor	0.838	0.779	0.814	0.751	0.454	0.495	0.977	0.932
KGCN-Leak-ReLU Average	0.888	0.826	0.871	0.810	0.681	0.644	0.978	0.933
KGCN-ELU Average	0.881	0.813	0.871	0.809	0.709	0.645	0.977	0.932
KGCN-ReLU Average	0.887	0.827	0.870	0.809	0.664	0.604	0.977	0.932

Among the three original KGCN aggregators, KGCN Sum and KGCN Concat perform significantly better than KGCN-Neighbour, and KGCN Neighbour shows a clear gap on 30-Core setting dataset; that may be because the KGCN Neighbour only aggregated the neighborhood's representation and does not include the information from the entity itself.

The last three lines of Table 3 represent the average performance of each non-linear function implementing on the three aggregators. We can see that Leaky ReLU function obtains the best results for all three datasets, which may be because Leaky ReLU overcomes the dead ReLU problem. When the representation values are negative, the Leaky ReLU returns a small fraction;in contrast, the ReLU always return 0. ELU function performs the worst among the three functions, which may be because for ELU, $\sigma(x) = x$ when $x > 0$ and $a \cdot (e^{x} - 1)$ when $x < 0$, KGCN is lack of ability to learn and update the value of a when the input value is negative.

We conducted experiments on how hyper parameter influences the performance of a KGCN-Leaky ReLU-Sum model. In Table 4, we can see the influence of the neighbor sampling size from 2 to 64. The model performs best when S=8. This is because a too small S does not have enough capacity to incorporate neighborhood information, while a too large S is prone to be misled by noises.

Table 4. AUC Result of Leaky ReLU-Sum KGCN with Different Neighbour Sampling Size S

S	2	4	8	16	32	64
10-Core	0.886	0.895	0.913	0.910	0.912	0.912
20-Core	0.872	0.886	0.903	0.903	0.899	0.892
30-Core	0.770	0.764	0.769	0.757	0.792	0.789

We can see the influence of depth of receptive field H on the model by setting H from 1 to 4 in Table 5, which illustrates that the model reacts sensitively on the variation of H.

The model performs best when H=2 and collapse dramatically when H is greater than 2, which may be because a larger H brings massive noises to the model and the too-long relation-chain makes little sense when inferring inter-item similarities.

Table 5. AUC Result of Leaky ReLU-Sum KGCN with Different Depth of Receptive Field H

H	**1**	**2**	**3**	**4**
10-Core	**0.878**	**0.904**	**0.504**	**0.505**
20-Core	**0.875**	**0.897**	**0.507**	**0.507**
30-Core	**0.785**	**0.770**	**0.382**	**0.385**

Table 6 displays the effects of dimension of embedding d on performance of the model. The result is intuitive: performance improves dramatically as the d increases; when d is 128, the model achieves the best performance, since the larger d can include more information of users and entities. When d is greater than 128, the model is drawn back by overfitting.

Table 6. AUC result of KGCN with different dimension of embedding D

D	**4**	**8**	**16**	**32**	**64**	**128**	**256**
10-Core	**0.820**	**0.858**	**0.893**	**0.900**	**0.903**	**0.914**	**0.903**
20-Core	**0.824**	**0.827**	**0.886**	**0.892**	**0.890**	**0.894**	**0.898**
30-Core	**0.780**	**0.817**	**0.806**	**0.753**	**0.751**	**0.785**	**0.773**

6. CONCLUSION AND FUTURE WORKS

In this paper, we proposed an efficient and effective data preprocessing and knowledge graph generation tool, Knowledge Graph Processor (KGP). By using the KGP, we constructed a knowledge graph for a Yelp dataset. Our proposed KGP can process JSON, CSV, and text files. More features could be supported, such as automatically extracting information from well-build knowledge graph database.

By testing KGCN on the Yelp knowledge graph and MovieLens-20M dataset with Leaky ReLU, ELU, and ReLU non-linear functions, Leaky ReLU is able to improve performance of the original KGCN for recommendation systems. The reason is because Leaky ReLU has an advantage on overcoming dead ReLU problem as well as its robust performance when the input values are negative.

We point out two avenues for future work: 1) The knowledge graph's content and quality can significantly affect the performance of KGCN. An interesting direction of future research is to quantify the quality of the knowledge graph dataset; 2) We investigated the influence of the nonlinear function in the first layer of the aggregator. Future work could explore the impact of the different nonlinear functions on the second layer, and the impact of the optimizer is also a valuable direction to study.

REFERENCES

[1] Jeannie Dougherty. (2019). Internet growth + usage stats 2019: Time online, devices, users. https://www.clickz.com/internet-growth-usage-stats-2019-time-online-devices-users/235102/

[2] Charlotte Johnson. (2019) How much Data is Produced every Day 2019? https://www.the-next-tech.com/blockchain-technology/how-much-data-is-produced-every-day-2019/

[3] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. (2018). "TEM: Tree-enhanced Embedding Model for Explainable Recommendation", WWW, ppl 1543–1552

[4] Yehuda Koren, Robert Bell and Chris Volinsky. (2009, August). Matrix Factorization Techniques for Recommender Systems. IEEE 2009

[5] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. (2011, July). Fast context-aware recommendations with factorization machines. SIGIR'11.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu. (2016). "Wide & Deep Learning for Recommender Systems", DLRS@RecSys. ppl 7–10.

[7] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. (2017)." Non-Local Neural Networks", arXiv preprint arXiv:1711.07971, vol. 10.

[8] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. (2018). "Graph attention networks", The 6th International Conferences on Learning Representations

[9] Yujia Li, Daniel Tarlow, Marc Brockschmidt, Richard Zemel. (2016). "Gated Graph Sequence Neural Networks", arXiv: Learning, 2016.

[10] Xiao Huang, Jingyuan Zhang, Dingcheng Li, Ping Li. (2019). "Knowledge Graph Embedding Based Question Answering", WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (January), ppl 105–113. https://doi.org/10.1145/3289600.3290956

[11] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. (2018). "Leveraging Metapath Based Context for Top-N Recommendation With A Neural Co-Attention Model", the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, ppl 1531–1540

[12] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. (2018). "Recurrent knowledge graph embedding for effective recommendation", the 12th ACM Conference on Recommender Systems. ACM, 297–305

[13] Joan Bruna, Wojciech Zaremba, Arthur Szlam, Yann LeCun. (2014). "Spectral Networks and Locally Connected Networks on Graphs", arXiv:1312.6203v3

[14] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo, (2019, May). "Knowledge Graph Convolutional Networks for Recommender Systems", WWW. ACM ISBN 978-1-4503-6674-8/19/05

[15] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo, (2018). "RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems", CIKM. ppl 417–426

[16] Crowd Flower. (2016). 2016 Data science report, https://visit.figure-eight.com/rs/416-ZBE142/images/ CrowdFlower_DataScienceReport_2016.pdf

[17] Amit Singhal. (2012). "Introducing The Knowledge Graph: Things, Not Strings," https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

[18] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. (2011). "Knowledge-based weak supervision for information extraction of overlapping relations," Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, ppl 541–550

[19] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. (2018). "Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks", the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 505–514

[20] Sherzod Hakimov, Sherzod Hakimov, Salih Atilay Oto and Erdogan Dogdu (2012). "Named entity disambiguation using linked data," Proc. 9th Extended Semantic Web Conf.

[21] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. (2014). "Personalized entity recommendation: A heterogeneous information network approach", the 7th ACM International Conference on Web Search and Data Mining. ACM, ppl 283–292.

[22] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. "Metagraph based recommendation fusion over heterogeneous information networks", the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, ppl 635–644.

AUTHORS

Xing Wei is a graduate students in the department of Computer Science at Lamar University. His research interests focus on data analytics for recommender systems.

Jiangjiang (Jane) Liu is a professor in the department of Computer Science at Lamar University. Her research interests are on data analytics, cloud computing, and high performance computing.

MALICIOUS NODE DETECTION IN SMART GRID NETWORKS

Faisal Y Al Yahmadi and Muhammad R Ahmed

[1]Marine Engineering Department, Military Technological College,
Muscat, Sultanate of Oman

ABSTRACT

Many countries around the world are implementing smart grids and smart meters. Malicious users that have moderate level of computer knowledge can manipulate smart meters and launch cyber-attacks. This poses cyber threats to network operators and government security. In order to reduce the number of electricity theft cases, companies need to develop preventive and protective methods to minimize the losses from this issue. In this paper, we propose a model based on software that detects malicious nodes in a smart grid network. The model collects data (electricity consumption/electric bill) from the nodes and compares it with previously obtained data. Support Vector Machine (SVM) model is implemented to classify nodes into good or malicious nodes by (high dimensional) giving the statues of 1 for good nodes and status of -1 for malicious (abnormal) nodes. The detection model also displays the network graphically as well as the data table. Moreover, this model displays the detection error in each cycle. It has a very low false alarm rate (2%) and a high detection rate as high as (98%). Future developments can trace the attack origin to eliminate or block the attack source minimizing losses before human control arrives.

KEYWORDS

Smart Grid Networks, Security, Malicious, Attacks, Support Vector Machine.

1. INTRODUCTION

Smart Grid Network (SGN) is an advanced network that merged new technologies and developed infrastructure to prepare the world to overcome the arising challenges expected to be faced in the coming decades. New implementations such as integration of alternative energy sources and decentralized generation will help overcome the growing global power demand expected with the adaptation of Electric vehicles EVs and other smart household appliances. SGN implementation of new technologies allows for two-way stream of both power and data [1]. These implementations will grant the network a greater ability to detect, react and pro-act towards power usage or other businesses. Suspicions power usage patterns by consumers will also be recognised and responded to with the new technology implementation. SGN enables service providers to monitor the behaviour of all stockholders of the electricity. SGN has the capability of enabling the consumer to become an active participant in the network. In order to ensure network economic feasibility and a high quality service with minimum losses, security and safety of supply is prioritised. Some of the benefits that SGN grants beneficiaries are as follows:

- Integration of alternative energy sources
- Decentralized generation
- Reliably electrical supply
- Greener power production
- Active consumer participation
- Better resilience towards grid blackouts

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 201-210, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110716

SGN implementation of information and communication technologies (ICT) allowed the new network to monitor, operate and control the system with added features. These control manners were only available for service providers at the generation phase. However, ICT helped to extend these manners across all SGN phases reaching transmission and distribution phases [2]. Two-way communication enables both service providers and companies to utilize the developed infrastructure for a more efficient grid. Two-way communication also allows consumers to be true active participants with ability to choose new power usage patterns that were not possible with the conventional grid. Moreover, to standardize the new SGN operation, National Institute of Standards and Technology (NIST) proposed an SGN model standardizing SGN architecture as shown in Fig. 1 below.

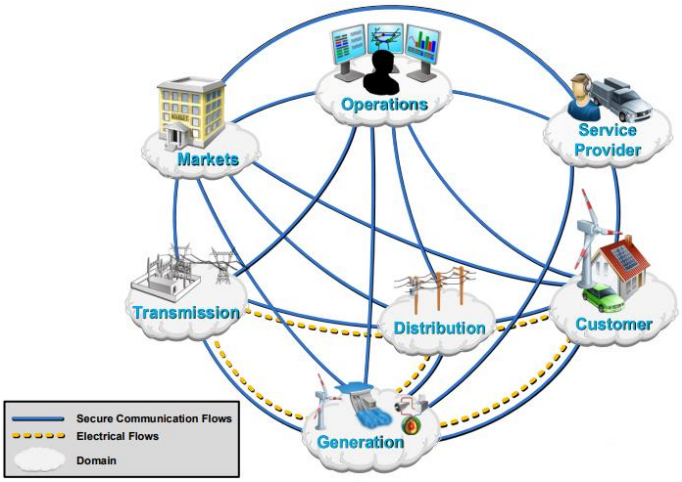


Fig. 1. Smart grid architecture model (SGAM) by [NIST] [3]

The proposed model lists seven domains, which are:

- Generation
- Transmission
- Distribution
- Operations
- Service providers
- Markets
- Consumer.

These above mentioned domains use secure communication in order to operate SGN efficiently. The proposed model also illustrates the electricity path between different domains, which are transmission, distribution, customer and generation, while communication flows across all seven domains.

Security provisioning is a critical necessity for any wired and wireless communication network [4]. Therefore, a machine-learning model will be adopted to detect attacks on SGN. Machine learning technology uses machines learning algorithms to artificially improve their performance as more data is being trained [5]. Machine learning has different techniques and models developed for various applications; one of the uses is solving classification problems. Support Vector Machine (SVM) is a classic machine learning technique which has the ability to classify high dimensional data [6]. This paper aims to develop an algorithm using one of the machine learning techniques, an SVM based model is used and simulated by MATLAB software. The simulation platform was chosen as MATLAB because it has the ability to classify attacked nodes by comparing collected data with average data collected from the same consumer/household. Attack detection revolves around two pillars, which are average electrical power consumption of the consumer (monthly/annually) and average (monthly/annually) electrical bill of the consumer. SGN is developed based on the wireless sensor network (WSN) concept. The data collecting

process starts with nodes representing consumers sending data to a central node and back to the supplier –witch in this case represents the developed algorithm-as shown in Fig. 2 below.

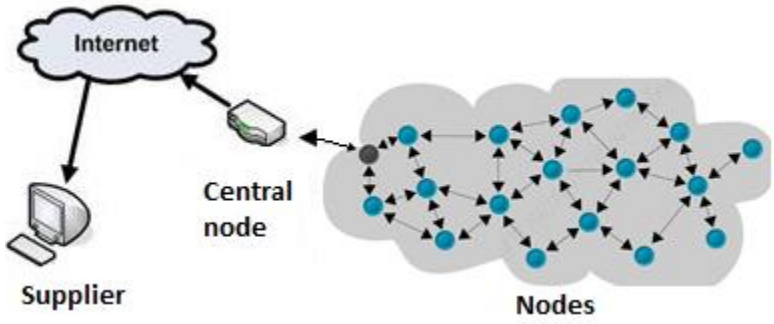


Fig. 2. A conceptual illustration of a generic WSN [7]

Considering the critical importance of SGN security, many algorithms have been developed to ensure secure functionality of SGN.

This paper is organized as follows. Section 2 contains an overview of related works followed by SGN implementations and assumptions in section 3. A detailed methodology is elaborated in section 4 with numerical results of the algorithm in section 5. The paper conclusion is in section 6.

2. RELATED WORKS

Smart grid network introduces enhancements and improved capabilities to the conventional power network making it more complex and vulnerable to different types of malicious attacks. Till today, several works have been done by many researchers to find the best way to detect malicious attacks but very few were focusing on the smart grid malicious attacks. Moreover, no significant importance has been given to finding the malicious attack based on the misbehaviour or abnormal behaviour of the node. Even though some researchers worked based on the misbehaviour but their main focus was to prevent or protect the routing. In the following section, related researchers work will be discussed:

Takiddin et al. in [8] provided answers to three major questions pertaining to the performance of electricity theft detectors in the presence of data poisoning attacks. By proposing a sequential ensemble detector based on a deep autoencoder with attention (AEA), gated recurrent units (GRUs), and feed forward neural networks. The proposed robust detector retains a stable detection performance that is deteriorated only by 1−3% in the presence of strong data poisoning attacks. However, in this method it is normally ensemble performs multiple learners, as a result computation get complicated, which reduce the speed and memory requirements rise.

Zhang et al. in [9] proposed a time series anomaly detection model based on the periodic extraction method of discrete Fourier transform. The detection model determines the sequence position of each element in the period by periodic overlapping mapping, thereby accurately describing the timing relationship between each network message. The experiments demonstrate that the model has the ability to detect cyber attacks such as man-in-the-middle, malicious injection, and Dos in a highly periodic network. The detection model also has a good anomaly detection capability. This model focus on the DoS attacks.

Jiang and Qian in [10] discussed defense mechanisms to either protect the system from attackers in advance or detect the existence of data injection attacks to improve the smart grid security. Focusing on signal processing techniques, this article introduces an adaptive scheme on detection of injected bad data at the control center. Jiang and Qian presented a detection scheme that can self-adaptively detect both non-stealthy and stealthy attacks. The scheme comprises determining two estimates of the state of the monitored system using the state measurement data provided by the remote sensing system at two sequential data collection slots, and determining bad data injection attacks by monitoring the measurement variations and state changes between the two slots. Analysis and simulation results shows that the proposed scheme is efficient in terms of data attack classification and detection accuracy. The research is good to detect data injection attacks.

Zhe et al. in [11] proposed a model based on machine learning to detect smart grid DoS attacks. The model collects network data, then selects features and uses PCA for data dimensionality reduction, and finally uses SVM algorithm for abnormality detection. By testing the SVM, Decision Tree and Naive Bayesian Network classification algorithms on the KDD99 dataset, it is found that the SVM model works best. This method has higher classification detection rate and accuracy, which can effectively improve the security of the smart grid DoS intrusion detection system. This method the data need to go thorough standardization process and in PCA we need to select the principle components otherwise it may miss data features.

Xia et al. in [12] suggest a method to identify all malicious users in a neighbourhood area network. The method uses Group Testing based Heuristic Inspection (GTHI) algorithm, which can estimate the ratio of malicious users on-line, mainly by collecting the information that how many malicious users have been identified during the inspection process. Based upon the ratio of malicious users, the GTHI algorithm adaptively adjusts inspection strategies between an individual inspection strategy and a group testing strategy. The GTHI algorithm outperforms existing methods in some aspects: compared with the BCGI algorithm, it has a wider range of applications; compared with the ATI algorithm, it can locate malicious users within much shorter detection time, regardless of the ratio of malicious users. However, this method does not include the user estimation in the testing phase.

Nandanoori et al. in [13] proposed a Koopman mode decomposition (KMD) based algorithm to detect and identify false data attacks in realtime. The Koopman modes (KMs) are capable of capturing the nonlinear modes of oscillation in the transient dynamics of the power networks and reveal the spatial embedding of both natural and anomalous modes of oscillations in the sensor measurements. The Koopman-based spatio-temporal nonlinear modal analysis is used to filter out the false data injected by an attacker. This algorithm detects the induced attack within 1 second of attack initiation in the presence of load changes in the network. This method normally works only work based on the false data injection.

Patil and Sankpal in [14] proposes an enhanced grid sensor placement (EGSP) algorithm to place grid sensors in the distribution network to monitor and control the smart meters installed in the field. The algorithm provides a simple and efficient way to place grid sensors in the distribution network for monitoring and controlling the smart meters deployed in the distribution network. A simulation model of distribution network has been developed for the analysis of the proposed algorithm. The analytical computation and simulation result shows that the number of grid sensors needed to track all the smart meters connected in the distribution network varies between half the number of SM nodes to equal number of SM nodes depending on how many SM nodes are connected to each EP node. In this method the computation is higher.

He et al. in [15] exploits a deep learning techniques to recognize the behavior features of FDI attacks with the historical measurement data and employ the captured features to detect the FDI

attacks in real-time. The proposed detection mechanism effectively relaxes the assumptions on the potential attack scenarios and achieves high accuracy. Furthermore, an optimization model is proposed to characterize the behavior of one type of FDI attack that compromises the limited number of state measurements of the power system for electricity theft. Method simulation results showed that the detection method can achieve high detection accuracy in the presence of the occasional operation faults. This work well only to predict the potential attack can happen.

The existing literature depicts that the vast majority of present methodologies to find the malicious in smart grid exists are in a general sense based on cryptographic primitives. Typically, in cryptographic solutions, the source uses cryptographic information to create and send additional authentication. As a results the extra information needed and the malicious can be detected based on the additional information data. The other introduced strategies are typically relying upon calculations and high level of training data. However, these methods have high computational overhead, because of every validation requires an immense number of checking to come up with the final decision about the malicious. Therefore, it is essential to develop an effective method to detect the malicious in the smart grid networks.

3. METHODOLOGY

Machine learning has many techniques, Support Vector Machines (SVM) based algorithm is used because of the model ability to classify unreliable data [16]. Which is suitable for high-dimensional data collected from across SGN. Therefore, SVM has been chosen for the proposed solution in this paper.

SVM model categorize the collected data by finding the optimal hyperplane shown in Figure 3 below, which will consist of the largest distance between the two different classes and that distance is called margin [17]. Margin is calculated from the nearest vector to the hyperplane and it must be without interior point as shown in Figure 3 below.

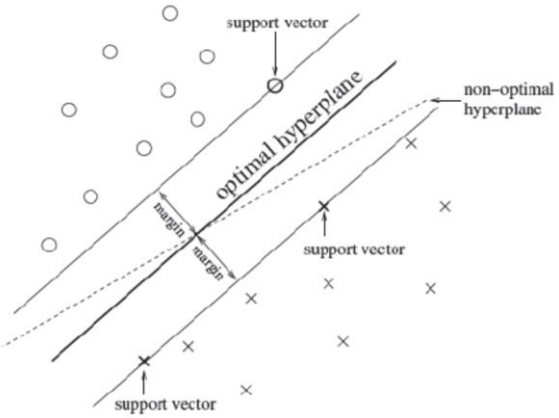


Fig. 3. SVM with its Optimal and non-optimal Hyperplane [18]

The closest point to the hyperplane which will be in contact with the margin parallel lines are called support vectors. Support vectors sets the hyperplane boundary [19]. Figure 3 also shows the two types of data, which are ×'s defining points of a value of 1 and O's defining points of a value of -1. The desired algorithm, a training phase to the system must be conducted offline using a resourceful information source. The training phase uses three Open System Intercommunication9 (OSI) layers, which are a physical layer followed by medium accessed control layer (MAC) ending with a network layer. After training then collecting the desired data,

a data trimming procedure will be implemented on these data sets. Data trimming is a vital step in order to reduce data size which will ultimately allow SVM to process it further. After completing data training and having training sets ready, classification can be done by a linear plane as illustrated in Fig. 4 below.

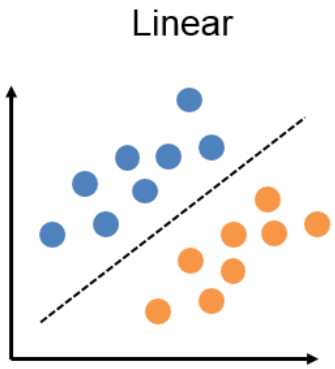


Fig. 4. Linear classification [20]

However, linear classification has limitations when it comes to classifying unreliable data [21]. Therefore, moving the data to a higher dimensional space will allow more functions that were not possible to be applicable such as mapping training sets.

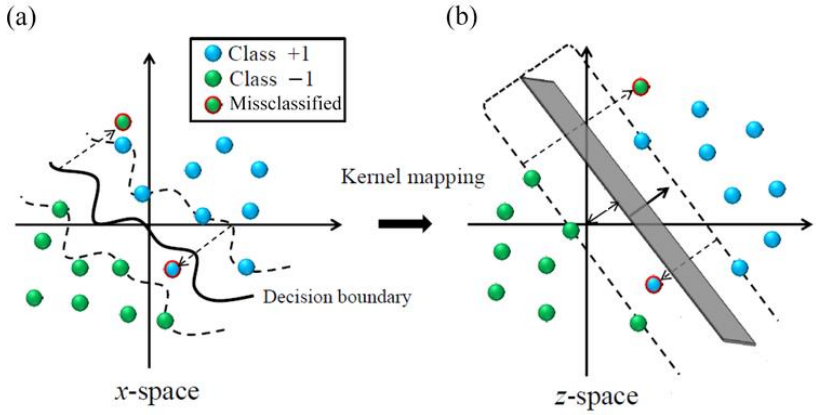


Fig. 5. A problem solved by mapping the training set [22]

As figure 5 shows, a problem that was unsolvable by using linear classification can be classified if training set data moved to a higher dimensional space. After understanding the theoretical part, it is now possible to explain the mathematical calculations behind the SVM method.

Assume that linear separability sample set is (x_i, y_i) with training data sets of:

$$i = 1, \dots, n, x \in R^d, y \in \{+1, -1\}$$

During this research, it's assumed that {1} is the normal and {-1} is the attacked or abnormal. Which leads to the equation of hyperplane classification as follows:

$$w.x + b = 0 \quad \text{---------- (1)}$$

In equation (1), the vector w is a normal vector while b is offset value. Initially we consider that the all the node is good and normal node. The best classifying hyperplane is supported by training data samples. While having this statement in mind, support vectors can be considered as the hyperplane training samples. Moreover, the formulation of the problem will be as follows:

$$\min \emptyset(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2}(w.w)$$
$$subject\ to\ y_i[(w.x_i) + b] - 1 \geq 0, i = 1,2,\dots,n \quad \text{------(2)}$$

Hence, a formulation of the classification function will be as follows:

$$f(x) = sgn\{(w^*.x) + b\} = sgn\{\sum_{i=1}^{n} a_i^* y_i (x_i.x) + b^* \quad \text{--- (3)}$$

And a formulation of the optimal classification function will be as follows:

$$f(x) = sgn\{\sum_{i=1}^{n} a_i^* y_i k(x_i.x) + b^* \quad \text{---------- (4)}$$

The function mentioned above $f(x)$ is kernel function while ai are function multipliers.
In our implementation, the nodes are connected to each other. Specifically, a node connects to a single neighbor node. When all nodes are connected, the optimal hyperplane will be calculated through the previously explained functions and all data from the nodes will be classified into either a normal node or attacked/abnormal node. This process is possible with the use of SVM because of the method ability to classify high-dimensional data.

4. SGN ASSUMPTIONS AND IMPLEMENTATION SCENARIO

In this paper, we considered the following assumptions to implement the methodology:

1- The end used will specify the area of interest. Area of interest has been modelled as a grid Ω of Nx × Ny points scenario. The specified area is given as $A = Nx \times Ny$. Where Nx is the area length in meters (X-Axis) and Ny is the width in meters (Y-Axis) giving the product of the area A.
2- Nodes are sensors that are stationary after deployment (generation of network) and it can be said that nodes are the smart meters that are located in all consumers participating in SGN. Nodes are the communication channel between service provider and consumers and are responsible for collecting and forwarding the monitored data to the central node illustrated previously in Fig. 2.
3- Nodes communicate with Neighbour nodes in a pre-set radio range of (0.25 m2) and to the central node.
4- SVM based algorithm is responsible for classification of nodes.
5- The network is assumed to be synchronized.

The hypothetical scenario was considered from one of the village –AFI- in Al Batinah South Governorate, sultanate of Oman. The Area A in the simulation was set by default to N_x of 500 (m) and a N_y of 500 (m) and The default setting of 75 nodes represents smart meters in households in the shown area above. Average electricity consumption set by default to 30 Kwh. Data collected from electricity provider [23] in the area mention above. The monthly bill is also set to a default 250 Omani Riyals calculated using the online bill calculated provided by the service provider [24].

5. RESULTS

The method was simulated based on the hypothetical scenario considered for the implementation. In order to create the scenario, we have obtained the data about the average electricity consumption of the inheritance of the subscribers from the electricity supplier [23] [24]. The Average electricity bill was set as a base to simulate the network. In our simulation, the basic parameter was set are as follows:

Table 1. Parameters

Parameters (components)	Used values
Number of nodes	75 node
Number of central nodes	1 node
Average electricity consumption	30 Kwh
Average monthly electric bill	250 OMR

In the evaluation process for the effectiveness of the implemented model, we have considered a set of matrices to determine the detection of the attacks.

a) Detection Rate: This is the detection percentage of the attacks based on the total number of attack was performed
b) False positive rate (false alarms): This is the ratio between the number classified as an abnormal node (which is considered as an attacked node) on the total number of normal connections.

The simulation in MATLAB gave us the attack detection accuracy of 98% and the False alarms rate as low as 2% from the total number of attacks. The simulation result is in the figures 7 and 8 shows the generated network and the distribution of the nodes. Fig 8 is the results when we detect the malicious based on the algorithm implemented.

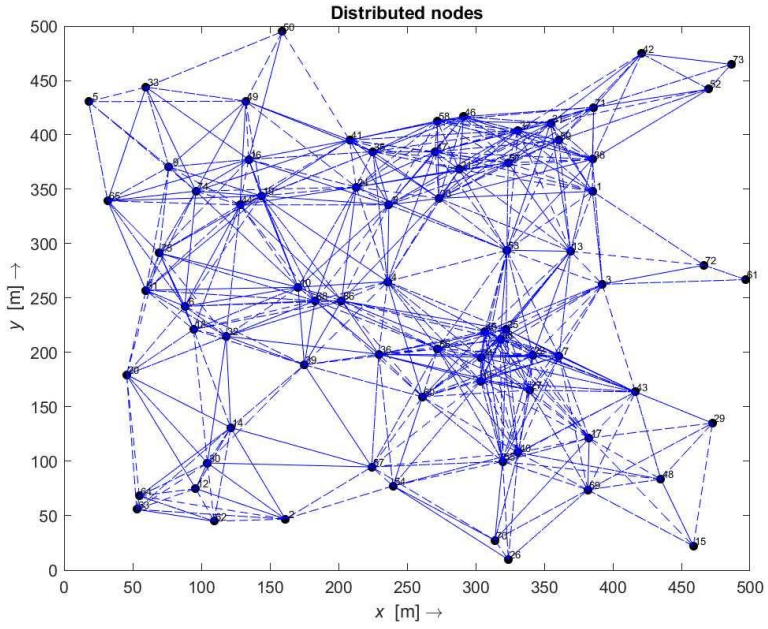


Fig. 7. The SGN Network

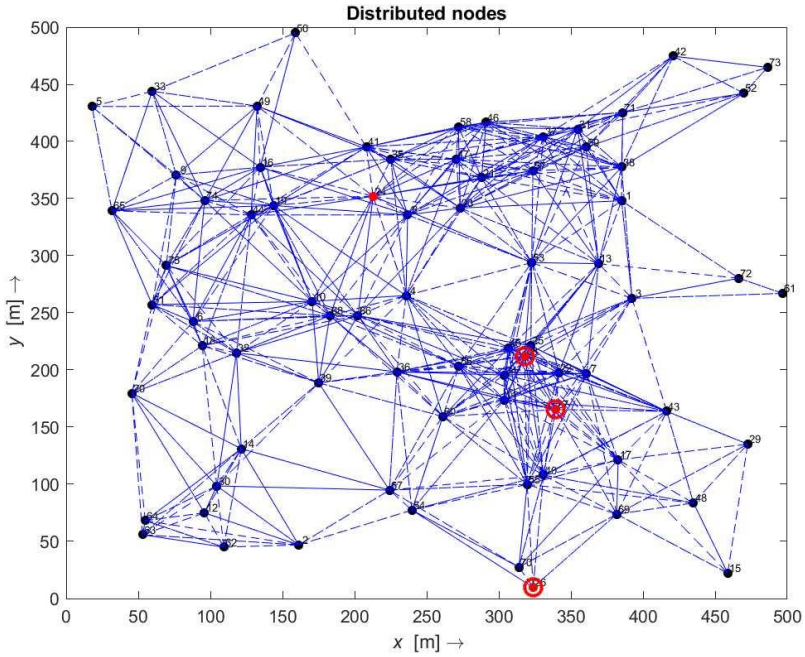


Fig. 8. Detection Malicious Nodes

In the Fig. 8, we can see that there are 75 nodes was distributed in the area and it was randomly checked. Based on the specified parameter the simulation results found 4 malicious nodes. The malicious nodes as circled red.

6. CONCLUSION

Smart Grid Network is backbone infrastructure is the information and communication technology that makes the network vulnerable to malicious attacks. It is essential to detect the attack work on the attack for the uninterrupted and effective supply of the electricity and generate the accurate bill. In this paper, the machine learning approach has been adopted. SVM makes memory efficient and effective for high dimension data. Considering this SVM-based classification framework of machine learning is implemented to detect misbehaving malicious nodes in smart grid networks. The simulation result in MATLAB gave us an effective detection outcome. The result shows us that our detection rate is about 90% and the false positive is only 2%. In future, we would like to simulate the network on a larger scale and implement it at the hardware level.

REFERENCES

[1] J. B. Ekanayake, N. Jenkins, K. Liyanage, J. Wu, and A. Yokoyama, *Smart Grid: Technology and Applications*. John Wiley & Sons, 2012.

[2] F. Skopik and P. Smith, *Smart Grid Security: Innovative Solutions for a Modernized Grid*. Elsevier Science & Technology Books, 2015.

[3] C. Greer *et al.*, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0," National Institute of Standards and Technology, NIST SP 1108r3, Oct. 2014. doi: 10.6028/NIST.SP.1108r3.

[4] M. R. Ahmed, S. M. Tahsien, M. Aseeri, and M. S. Kaiser, "Malicious attack detection in underwater wireless sensor network," in *2015 IEEE International Conference on Telecommunications and Photonics (ICTP)*, Dhaka, Bangladesh, Dec. 2015, pp. 1–5, doi: 10.1109/ICTP.2015.7427952.

[5] P. Langley, *Elements of Machine Learning*. Morgan Kaufmann, 1996.

[6] S. Suthaharan, "Support Vector Machine," in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, S. Suthaharan, Ed. Boston, MA: Springer US, 2016, p. 1.
[7] "Fig 1: A conceptual illustration of a generic WSN," *ResearchGate*. https://www.researchgate.net/figure/A-conceptual-illustration-of-a-generic-WSN_fig1_301241534 (accessed Jan. 13, 2021).
[8] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust Electricity Theft Detection Against Data Poisoning Attacks in Smart Grids," *IEEE Transactions on Smart Grid*, pp. 1–1, 2020, doi: 10.1109/TSG.2020.3047864.
[9] L. Zhang, X. Shen, F. Zhang, M. Ren, B. Ge, and B. Li, "Anomaly Detection for Power Grid Based on Time Series Model," in *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Aug. 2019, pp. 188–192, doi: 10.1109/CSE/EUC.2019.00044.
[10] J. Jiang and Y. Qian, "Defense Mechanisms against Data Injection Attacks in Smart Grid Networks," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 76–82, Oct. 2017, doi: 10.1109/MCOM.2017.1700180.
[11] W. Zhe, C. Wei, and L. Chunlin, "DoS attack detection model of smart grid based on machine learning method," in *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Jul. 2020, pp. 735–738, doi: 10.1109/ICPICS50287.2020.9202401.
[12] X. Xia, Y. Xiao, W. Liang, and M. Zheng, "GTHI: A Heuristic Algorithm to Detect Malicious Users in Smart Grids," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 805–816, Apr. 2020, doi: 10.1109/TNSE.2018.2855139.
[13] S. P. Nandanoori, S. Kundu, S. Pal, K. Agarwal, and S. Choudhury, "Model-Agnostic Algorithm for Real-Time Attack Identification in Power Grid using Koopman Modes," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Nov. 2020, pp. 1–6, doi: 10.1109/SmartGridComm47815.2020.9303022.
[14] Y. S. Patil and S. V. Sankpal, "EGSP: Enhanced Grid Sensor Placement Algorithm for Energy Theft Detection in Smart Grids," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, Mar. 2019, pp. 1–5, doi: 10.1109/I2CT45611.2019.9033759.
[15] Y. He, G. J. Mendis, and J. Wei, "Real-Time Detection of False Data Injection Attacks in Smart Grid: A Deep Learning-Based Intelligent Mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017, doi: 10.1109/TSG.2017.2703842.
[16] N. Cristianini, J. Shawe-Taylor, and D. of C. S. R. H. J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
[17] S. S. Keerthi and C.-J. Lin, "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, Jul. 2003, doi: 10.1162/089976603321891855.
[18] "Fig. 4. SVM classification with a hyperplane that maximizes the...," *ResearchGate*. https://www.researchgate.net/figure/SVM-classification-with-a-hyperplane-that-maximizes-the-separating-margin-between-the-two_fig3_221926953 (accessed Jan. 13, 2021).
[19] G. C. Calafiore and L. E. Ghaoui, *Optimization Models*. Cambridge University Press, 2014.
[20] J. Sullivan, "Neural Network from Scratch: Perceptron Linear Classifier," *John Sullivan*, Aug. 16, 2017. https://jtsulliv.github.io/perceptron/ (accessed Jan. 13, 2021).
[21] R. Grosse, "Lecture 3: Linear Classification," p. 10.
[22] "Figure 1. Graphical presentation of the support vector machine...," *ResearchGate*. https://www.researchgate.net/figure/Graphical-presentation-of-the-support-vector-machine-classifier-with-a-non-linear-kernel_fig1_299529384 (accessed Jan. 13, 2021).
[23] "Pages - Home." https://mzec.nama.om/en-us/Pages/home.aspx (accessed Mar. 19, 2021).
[24] "Bill Calculator." https://mzec.nama.om/en-us/Pages/billcalculator.aspx (accessed Mar. 19, 2021).

COMPARATIVE ANALYSIS OF QUALITY OF SERVICE SCHEDULING CLASSES IN MOBILE AD-HOC NETWORKS

Thulani Phakathi, Bukohwo Michael Esiefarienrhe
and Francis Lugayizi

Department of Computer Science, North-West University,
Mafikeng, South Africa

ABSTRACT

Quality of Service (QoS) is now regarded as a requirement for all networks in managing resources like bandwidth and avoidance of network impairments like packet loss, jitter, and delay. Media transfer or streaming would be virtually impossible if QoS parameters were not used even if the streaming protocols were perfectly designed. QoS Scheduling classes help in network traffic optimization and the priority management of packets. This paper presents an analysis of QoS scheduling classes using video traffic in a MANET. The main objective was to identify a scheduling class that provides better QoS for video streaming. A simulation was conducted using NetSim and results were analyzed according to throughput, jitter, and delay. The overall results showed that extended real-time Polling Service (ertPS) outperformed the other classes. ertPS has hybrid features of both real-time Polling Service (rtPS) and Unsolicited Grant Service(UGS) hence the enhanced performance. It is recommended that ertPS scheduling class should be used in MANET where QoS consideration is utmost particularly in multimedia streaming applications.

KEYWORDS

Routing protocols, MANETs, Scheduling, QoS in MANET, rtPS protocol.

1. INTRODUCTION

Quality of Service (QoS) is a widely used term associated with the measurement of the overall performance of a network service in multimedia applications. It is a necessary function within MANETs to ensure an end to end quality in terms of throughput, jitter, bandwidth, or delay. One of the biggest challenges in MANETs is the design of a secure and efficient routing protocol that will guarantee an acceptable level of QoS during the routing process as MANETs communicate only when they are in the range of each other or near a base station. There is no central controlling device. These design characteristics make it difficult to ensure QoS in the network.

The support for QoS [1] constantly requires link-state information to be forever present i.e. bandwidth, loss rate, error rate, and delay. Supporting the mentioned requirements is often a huge task as the quality of the ad-hoc link can abruptly change because nodes are mobile and operate in dynamic environments. Earlier approaches to QoS were based on the virtual circuit model which meant that there should be fixed connection management before communication so that for the duration of that session there would be a guarantee of route and reservation between the source node and the destination node. This was a great approach but it lacked the flexibility needed to dynamically adapt to mobile ad-hoc networks where the path and reservation needed to

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 211-220, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110717

respond dynamically and in real-time to the ever-changing topology and resource needs [2]. One of the threats to QoS support is the maintenance of an acceptable service level. By default, QoS support in MANETs is linked to the routing protocol's performance because the flow between the source nodes to the destination is not always straightforward. There is a high chance that within the lifetime of an on-going session, the flow gets rerouted and this happens frequently. When this happens, there is also a change in resource provisioning and needs as the flow would be now on a new path. The QoS agreement from the initial path may not be valid and so would be the assumption that the route and reservation remain fixed in that duration of a session. This is the reason why many researchers have been constantly coming up with mechanisms and frameworks that can support QoS in MANETs adaptively and can respond to its intrinsic behavior [1].

In MANETS, it is difficult to guarantee QoS than traditional networks because the wireless bandwidth is shared amongst nodes and the network topology is forever changing as nodes are in transit. It is for this reason that QoS provisioning can only be achieved through extensive collaboration among nodes for route establishment and securing the available network resources. The provision of QoS in a network can be classified into two namely soft and hard QoS approaches. In the soft QoS approach, the QoS requirements are not guaranteed for the whole session. This may be due to insufficient available network resources. Hard QoS approaches ensures the availability of network resources to meet all the QoS requirements of a connection and such requirements can be sustained for the entire duration of the session. QoS Provisioning improves end to end performance of nodes in heavily congested network scenarios through QoS aware routing, resource reservation, admission control, scheduling, and traffic analysis. Some of the other reasons why QoS provisioning is still a challenge in MANETs is because of limited resource availability, error-prone channel, lack of central controller, hidden terminal problem, insecure medium and dynamic network topology. This makes it hard to achieve and maintain end to end QoS. The most crucial system components to QoS provisioning in MANETs are network resources and their availability to process application data. There is no standardized mechanism to guarantee absolute QoS in MANETs but only some level of QoS can be achieved through different methods.

This work is arranged as follows: Section 2 discusses mechanisms that have been implemented in literature according to their various categories, section 3 presents the proposed framework in detail including the sinkhole attack implementation and the counterattack intrusion detection system. Section 4 presents the results and discussion and lastly, section 5 gives the conclusion and suggestion for future work.

1.1. IEEE 802.16 & QoS Scheduling Approaches

The IEEE 802.16e is an amended version of the IEEE 802.16 that was initially approved for targeted fixed applications. The amendment was enhanced for adding mobility support for nomadic and mobile applications. The new additions introduced modifications in the physical layer from OFDM to scalable OFDMA[3]. There were additional modifications within the MAC layer for resource management, roaming, security, and handoff. Additional enhancements in the MAC include sleep/idle-mode for mobile nodes, power-saving classes, paging, locating, and defining messages for handover procedures. The IEEE 802.16e specification adds a new scheduling service called extended real-time Polling Service (ertPS), which combines the efficiency of Unsolicited Grant Service (UGS) and real-time Polling Service (rtPS). It allows unsolicited bandwidth grants like UGS, but with dynamic size like rtPS. This yields a services class supporting real-time service flows with variable size data packets, suitable for Voice over IP (VoIP) with silence suppression[4].

1.1.1. QoS Scheduling Classes

The scheduling process determines the order in which packets in a queue should be processed. Priority scheduling involves using an algorithm that allows the router to fix the priority level for the packets that would be coming from different sources and directions. Higher priority packets are processed first and sent out.

A. Unsolicited Grant Service (UGS)

The UGS scheduling service type is designed to support real-time data streams consisting of fixed-sized data packets issued at periodic intervals, for example, Pulse Code Modulation(PCM) pone signals and Voice over IP without silence suppression. The base station provides data grants at fixed periodic intervals to reduce the latency and overhead of the subscriber stations. This class is used for high priority packets[5].

B. Real-time Policing Service (rtPS)

The rtPS scheduling service type is designed to support real-time data streams consisting of variable-sized data packets that are issued at periodic intervals. This would the case, for example, for MPEG (Moving Pictures Experts Group) video transmission. The base station provides periodic unicast requests that are in line with the flow's real-time needs and allow the subscriber stations to specify the size of the grant required. This service requires more resources than UGS because of the request overheads it requires but supports grant sizes of different sizes for optimum efficiency in real-time data transport[6].

C. Extended Real-time Polling Service (ertPS)

The ertPS is a scheduling mechanism that builds on the efficiency of both UGS and rtPS. UGS allocations are fixed in size, ertPS allocations are dynamic. The ertPS is suitable for variable-rate real-time applications that have data rate and delay requirements. Priority for packets is Normal [7].

D. Non-real-time Polling Service (nrtPS)

The nrtPS is designed to support delay-tolerant data streams consisting of variable-size data packets for which a minimum data rate is required. The standard considers that considered that this would be the case, for example, for an FTP transmission. The base station provides regular unicast uplink requests to guarantee request opportunities even during network congestion. A CID is a unique connection identifier assigned to every connection through the base station. *nrtPS* states that base station polls nrtPS CIDs at every second or less[8].

E. Best Effort(BE)

The BE service is designed to support data streams for which no minimum service guarantees are required and therefore may be handled on *bes*t basis.

2. RELATED WORK

The authors in [9 presented the use of two differing QoS level schemes; Best Effort level and High Effort level for the development of streams. Their proposed framework involves a center point that isolates streams with different level needs by dispensing them towards a similar goal. This was done using a procedure with two tables DRT (Dedicated Routing Table) and Standard

Routing Table (SRT) responsible for the movement of data streams. They proposed QOLSR (Quality Optimized Link State Routing) as the basic OLSR protocol was considered for development. This proposal did not consider extremely important aspects of QoS information exchange like reservation signaling, Connection Admission Control, and the stream classifier

Authors in [9] proposed scheduling strategies for nrtPS connections in IEEE 802.16 networks. The three strategies were categorized into dynamic,cross-layer, and conventional schedulers. The authors, in their review paper, additionally explored neuro-fuzzy scheduling and game theory-based techniques for further optimization techniques. They also highlighted that theories around the development of nrtPS are still at their growing stages and the scope is huge. They also suggested that one way to improve the *nrtPS* traffic class would be to use a technique that bases its scheduling decision based on the queue length of the nrtPS class. This would be through the determination of bandwidth requests by nrtPS and storing these requests in a virtual queue at the base station. The bandwidth requests are then sorted by the lowest queue length and each virtual queue is assigned a counter. The algorithm then does verification of the bandwidth requests made by the connection if whether they were satisfied or not. If the connection is not satisfied, then the algorithm checks the availability of more symbols to be allocated. If there are more symbols, then they are allocated to the connection and its counter will be decreased. The proposed algorithm tries to evade starvation of *nrtPS* connections. The proposed algorithm was intended for non-real-time applications and that excludes many high performing scheduling classes like UGS and ertPS

Authors in [10] presented a QoS aware framework to bridge the gap between security and QoS for the optimal functioning of a MANET. The study presented the existing challenges, attacks, and architectures as highlighted in the literature. They also presented a security keying system linked with the basic configuration of OLSR. Although this work used UGS as their preferred QoS scheduling class, the focus was on security and QoS attainment. The results with UGS showed better QoS than other scenarios.

Authors in [11] proposed a queue length scheduling algorithm for non-real-time and real-time traffic. The scheduling is initiated based on the number of MPDUs present at the start of the uplink subframe. The algorithm was designed to help in providing excess resources to non-real-time packets and also considering the queue length while guaranteeing QoS. The algorithm was not designed for real-time applications like video conferencing.

The authors in [12] proposed a method to improve the protocol Dual Busy Tone Multiple Access(DBTMA) using two elements called: busy tones and Ready To Send/Clear to Send (RTS/CTS) dialogues. A strategy to improve the effectiveness of fast retransmission was used. This retransmission strategy involved using a negative acknowledgment after a collision is caused by hidden nodes. The hidden node then listens to the negative acknowledgment signal to determine the requirement fast retransmission scheme. This method was compared with other methods in terms of their various network parameters and it showed improved QoS in terms of throughput, packet delivery ratio, and packet loss as compared to existing architectures but did not specifically speak to QoS scheduling.

Authors in [13] proposed a Medium Access Control (MAC) protocol that defines mechanisms for QoS provision and bandwidth allocation. The authors did not, however, specify the details on how to schedule traffic. This is necessary so that the vendor can differentiate their product through implementation. The authors introduce an efficient QoS scheduling strategy based on the distributed and hierarchical architecture for IEEE 802.16 systems. The simulation results provided positive feedback in terms of QoS guarantees for all types of traffic as defined by the IEEE 802.16 standard.

3. MATERIALS & METHODS

The research methodology used in this work is simulation as according to [14], simulation is defined as an experimentation method which involves the creation of an artificial environment within which relevant data can be generated. This is done in a controlled environment and it permits the observation of system dynamic behaviors. One of the purposes of simulation is to perform real decision making or diagnostic tasks. Decisions taken through analytical formulations are less preferred compared to simulations because they do not provide certainty. Network simulators best describe and represent the state of the network. These include nodes, links, switches, hubs, and routers. In modern-day simulators, they are either Graphical User Interface (GUI) driven or Command Line Interface (CLI) driven. Simulation is mainly used in performance analysis, comparison, or even management and also for determining how a network would behave in a real-life situation. The generated result of the simulation helps in identifying the performance attributes of the network. The flair to the entire simulation software is that there is a wide range of tools that ensure the generation of excellent results (GUI-based). There is a choice of network traffic selection, programming environments, projection, and statistical data representation. These are all part of the package of the simulation tool.

Discrete event simulation and experimentation using NetSim are implemented to adequately characterize variables, corresponding states, and events that change the value of these variable states in some rule-oriented but stochastic manner. The entities are the different components in the proposed framework.

In this study, the QoS measures used to ensure that quality is optimized in the network are throughput, delay, and jitter:

Throughput: It represents the number of bits forwarded from the Medium Access Control (MAC) to higher layers in all nodes in the network. It is measured in bits per second. The throughput may also be referred to as the average number of packets successfully transmitted or received per second. This work focused on the application throughput which is the total user data sent to the intended destination per second as shown in equation 1.

$$\text{Throughput} = \frac{P_d}{t} \quad (1)$$

Where *Pd* is the number of packets delivered and *t* is the time in seconds.

Delay: This is normally the time taken for one packet to be transmitted from the source node to the destination node. This performance metric evaluates the routing protocol's effectiveness in the use of network resources. Delay may be caused by several obstacles including transfer time, buffering during discovery latency, interference queue, and propagation and it is represented as shown in equation 2:

$$\text{Delay} = T_{rx} - T_{st} \quad (2)$$

Where T*rx* is the time the packet is received and T*st* is the time the packet is sent.

Jitter: This is the variation in time between route changes and data packets arriving. The variation may be caused by internal sources like data transmission errors, the presence of a malicious node, and network congestion. It usually affects the audio quality of the video if its level is high. The formula to calculate jitter is shown in equation 3:

$$\text{Jitter} = D_t - D_p \quad (3)$$

Where D*t* is the transmission delay of the current packet and D*p* transmission delay of the previous packet.

Table 1: Simulation parameters of the Network

Parameters	Value(s)
Simulator	NetSim Standard v12.1
Application Protocols	OLSR
Grid length	1000m*1000m
Simulation time	100seconds
Traffic type	Video conferencing
QoS Class	rtPS,ertPS,BE,nrtPS & UGS
Node movement model	Random Waypoint
Trajectory	Random
Transport Protocol	UDP
Speed	50km/h
Refresh interval	2s
Encryption Algorithm	Advanced Encryption Standard
Node density	10

4. RESULTS & DISCUSSION

The simulation results for the QoS running video-conferencing application for 10 nodes according to throughput, delay, and jitter:

4.1. Throughput

A. Best Effort, ertPS, NrtPS , rtPS & UGS

For Best Effort, the first 1000ms show a sharp increase in throughput levels up to approximately 0.35Mbps then continues to a nearly constant figure for the rest of the simulation time up until it reaches a peak of about 0.447Mbps after 7800ms. *Best effort* is considered low priority in terms of scheduling but gave an impressive throughput level for video streaming. For *ertPS*, the first 2000ms represents a refresh interval time before the application started. The next 18000ms shows a sharp rise in throughput levels up to 0.37Mbps as shown in fig.2. The highest recorded throughput is 0.458 at the end of the simulation time.

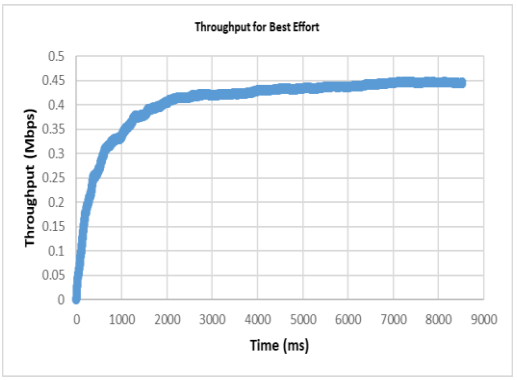


Fig.1: Throughput for Best Effort

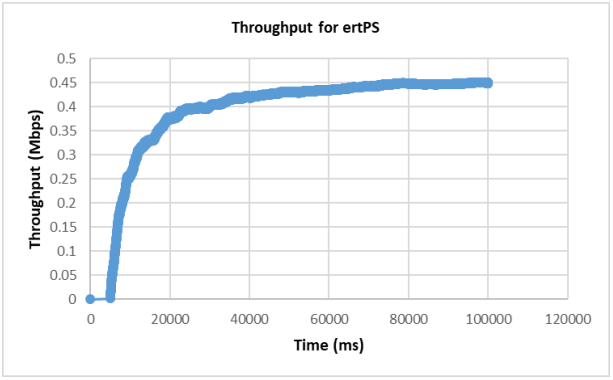


Fig.2: Throughput for ertPS

For *nrtPS*,as shown in Fig.3, the first 2000ms represents a refresh interval time before the application started. The next 18000ms shows a sharp rise in throughput levels up to 0.375Mbps. The highest recorded throughput is 0.45Mbps at the end of the simulation time. In fig.4, the first 2000ms represents a refresh interval time before the application started for *rtPS*. The highest recorded throughput is 0.451Mbps. In the first 18000ms of *UGS's* performance, there is a sharp rise in throughput levels up to 0.377Mbps as shown in fig.5. The highest recorded throughput is 0.45Mbps at the end of the simulation time. *UGS* is known for high priority scheduling performed averagely or rather less than expected in terms of throughput as compared to *nrtPS*. The throughput levels from all three scheduling algorithms are virtually similar.

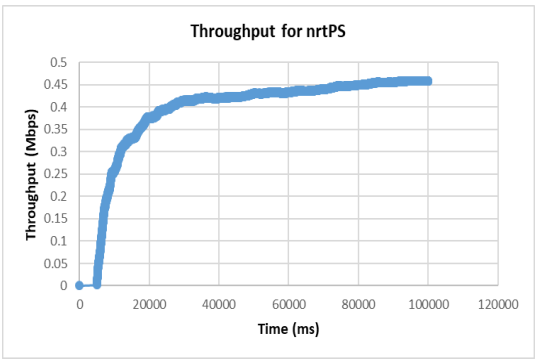


Fig.3: Throughput for nrtPS

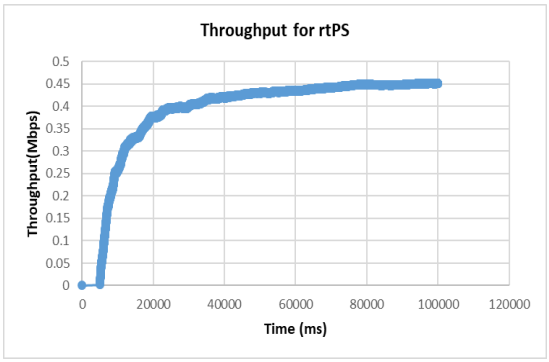


Fig.4: Throughput for rtPS

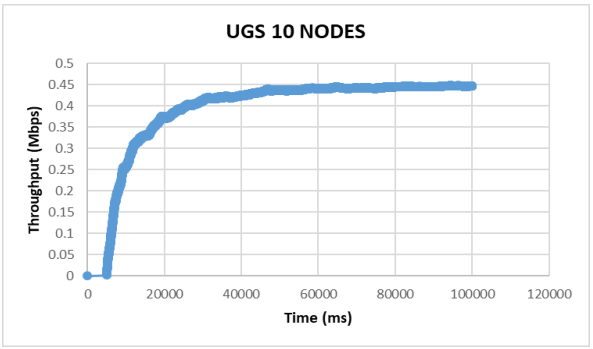


Fig. 5 : Throughput for UGS

4.2. Delay

According to Fig 6, the QoS classes portrayed a consistent delay for all 10 nodes. *Best Effort* obtained 6371.5 microsec and *ertPS* obtained 6262.1 microsec for all 10 nodes.*rtPS* had 6275.46microsec,ertPS obtained 6262.096 microsec and *nrtPS* obtained 6519.375 microsec for all 10 nodes. The performance of *UGS* in terms of delay is far better as compared to the other algorithms. *nrtPS* obtained the highest delay because it mostly works for non-real-time applications. *Best effort* also uses low priority scheduling for packets and this is not ideal for video streaming applications. An alternative to *UGS* would be *ertPS* and *rtPS* as they are accommodative to video streaming. The other scheduling algorithms are not weaker than the recommended ones but it depends on the application traffic that is being implemented.

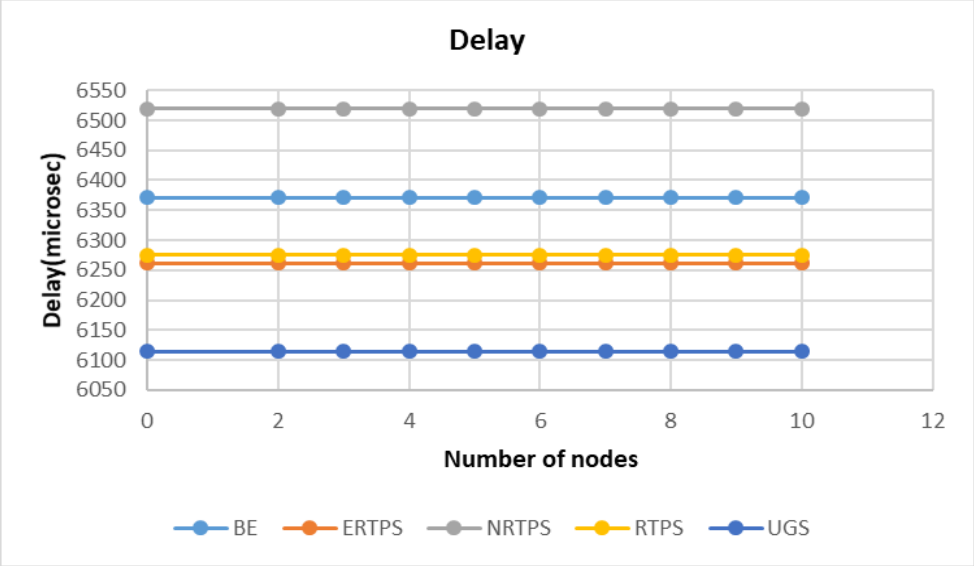


Fig.6 : Delay for all QoS classes

4.3. Jitter

The QoS classes have a consistent jitter level for all 10 nodes as shown in Fig 7. *Best Effort* obtained 2732.75 microsec and *ertPS* obtained 2663.96 microsec for all 10 nodes. *rtPS* had 2683.74 microsec, and *nrtPS* obtained 6519.375 microsec for all 10 nodes. The performance of *UGS* in terms of jitter is slightly higher than *ertPS*. *ertPS* builds on the efficiency of both UGS and RTP hence its excellent performance with the only difference in that its allocations are dynamic as compared to UGS whose allocations are fixed. *nrtPS* got the highest jitter because it mostly works for non-real-time applications and the application ran was in real-time. *Best effort* also uses low priority scheduling for packets and this is not ideal for video streaming applications hence its high jitter levels. An alternative to *ertPS* would be *UGS* and *rtPS* as they are accommodative to video streaming and share a few properties.

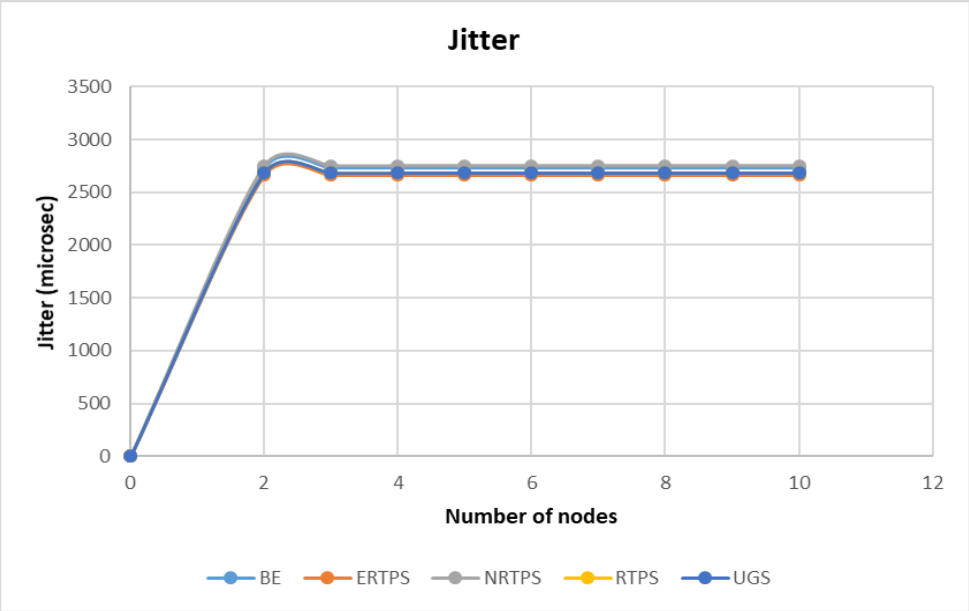


Fig.7: Jitter for all QoS classes

5. CONCLUSIONS

This paper presented and discussed an analysis of different QoS scheduling classes running video conferencing applications in MANETs. The comparative analysis aimed to draw a class that is best suitable for streaming purposes in a MANET in terms of QoS provision. The simulation results provided unique results in terms of the performance of these QoS classes. UGS, a high priority scheduling class, was outperformed in terms of throughput and jitter by ertPS. The only logical explanation for this unique result is that ertPS uses UGS and rtPS properties and that it capitalizes on its dynamic scheduling properties. As future work, the utmost intention is to further improve the performance of *ertPS* by analyzing the source code and making improvement of its parameters. The new source code will be an addition to improve the quality of the ertPS protocol for optimum performance. The advanced optimization techniques using ertPS protocol is another option that will also be considered.

ACKNOWLEDGEMENTS

This work would not be possible without the support from the Faculty of Natural and Agricultural Sciences, its Postgraduate office of the North-West University and our partners at TETCOS, India who provided the test-bed for our practical and evaluation of this work.

REFERENCES

[1] F. A. Khan, M. Imran, H. Abbas, and M. H. Durad, "A detection and prevention system against collaborative attacks in Mobile Ad hoc Networks," *Future Generation Computer Systems*, vol. 68, pp. 416-427, 2017.

[2] D. Hurley-Smith, J. Wetherall, and A. Adekunle, "SUPERMAN: Security Using Pre-Existing Routing for Mobile Ad hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 2927-2940, 2017.

[3] S. Benkirane and M. Benaziz, "Performance evaluation of ieee 802.11 p and ieee 802.16 e for vehicular ad hoc networks using simulation tools," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 2018, pp. 573-577.

[4] A. A. E. Abdalla, E. A. A. Mousa, and S. A. M. Ali, "Performance Evaluation of VoIP over WiMAX Network," AlMughtaribeen university, 2017.
[5] G. Singh, S. Kumar, and S. Malhotra, "A Survey on service flow support based networks."
[6] I. Almerhag and N. Aboalgasm, "The Effect of Mobility on the Performance of VoIP Application in WiMAX Networks."
[7] M. E. M. ABDALLA, "Performance Evaluation of VoIP QoS in WiMAX Networks," Sudan University of Science and Technology, 2017.
[8] B. D. Deebak, E. Ever, and F. Al-Turjman, "Analyzing enhanced real-time uplink scheduling algorithm in 3GPP LTE-advanced networks using multimedia systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, p. e3443, 2018.
[9] P. G. Akashdeep, "Analysis of Scheduling Strategies for Non-Real-Time Class in IEEE 802.16 Networks."
[10] T. Phakathi, F. Lugayizi, and M. Esiefarienrhe, "Quality of Service-aware Security Framework for Mobile Ad hoc Networks using Optimized Link State Routing Protocol," *arXiv preprint arXiv:2010.01852*, 2020.
[11] K. Raghu, S. K. Bose, and M. Ma, "Queue based scheduling for IEEE 802.16 wireless broadband," in *2007 6th International Conference on Information, Communications & Signal Processing*, 2007, pp. 1-5.
[12] M. Sivaram, V. Porkodi, A. S. Mohammed, V. Manikandan, and N. Yuvaraj, "Retransmission DBTMA protocol with fast retransmission strategy to improve the performance of MANETs," *IEEE Access*, vol. 7, pp. 85098-85109, 2019.
[13] J. Sun, Y. Yao, and H. Zhu, "Quality of service scheduling for 802.16 broadband wireless access systems," in *2006 IEEE 63rd Vehicular Technology Conference*, 2006, pp. 1221-1225.
[14] K. Dooley, "Simulation research methods," *Companion to organizations*, pp. 829-848, 2002.

Hierarchical Virtual Bitmaps for Spread Estimation in Traffic Measurement

Olufemi Odegbile[1], Chaoyi Ma[2], Shigang Chen[3], Dimitrios Melissourgos[4] and Haibo Wang[5]

Department of Computer and Information Science and Engineering
University of Florida, Gainesville, Florida, USA
Email: {[1]oodegbile, [2]ch.ma, [3]sgchen, [4]dmelissourgos, and [5]wanghaibo}@ufl.edu

Abstract. This paper introduces a hierarchical traffic model for spread measurement of network traffic flows. The hierarchical model, which aggregates lower level flows into higher-level flows in a hierarchical structure, will allow us to measure network traffic at different granularities at once to support diverse traffic analysis from a grand view to fine-grained details. The spread of a flow is the number of distinct elements (under measurement) in the flow, where the flow label (that identifies packets belonging to the flow) and the elements (which are defined based on application need) can be found in packet headers or payload. Traditional flow spread estimators are designed without hierarchical traffic modeling in mind, and incur high overhead when they are applied to each level of the traffic hierarchy. In this paper, we propose a new Hierarchical Virtual bitmap Estimator (HVE) that performs simultaneous multi-level traffic measurement, at the same cost of a traditional estimator, without degrading measurement accuracy. We implement the proposed solution and perform experiments based on real traffic traces. The experimental results demonstrate that HVE improves measurement throughput by 43% to 155%, thanks to the reduction of per-packet processing overhead. For small to medium flows, its measurement accuracy is largely similar to traditional estimators that work at one level at a time. For large aggregate and base flows, its accuracy is better, with up to 97% smaller error in our experiments.

1 Introduction

Traffic measurement is critical in supporting modern network functions [1], [2], [3], [4], [5], [6],[7]. Accurate information about current traffic loads is needed for routing optimization and load balancing among middleboxes that provide web proxing, firewalling and other functions [4], [5]. Network statistics are widely used to establish normal traffic patterns and detect anomalies that deviate from the normal [8]. Flow-level measurement provides fine-grained data to assess the behavior of individual hosts or subnets for performance or cybersecurity analysis [9]. While packet forwarding is the key function for any high-speed switch or router, auxiliary functions such as traffic measurement should be made as space-time efficient as possible, not only to avoid becoming a throughput bottleneck but also to save resources (e.g., cache memory and hardware circuits) for other important functions.

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 221-238, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110718

NetFlow [10] is a commonly used traffic measurement tool. It provides statistics such as number of packets and number of bytes for each TCP flow. A flow is a set of packets identified by common user-defined characteristics, such as source and destination IP addresses, source and destination ports, and protocol types. This paper extends the measurement function in two ways. First, we consider a hierarchical flow model, which we explain through an example where a cloud provider allocates vitual machines (VMs), racks of physical machines or whole pods to its clients, where each pod contains multiple racks. Suppose the provider wants to implement a measurement function at its datacenter gateway to monitor traffic between its clients and the Internet. Flows can be defined at the level of VMs, where each *VM flow* consists of all packets from a VM to the Internet (or from the Internet to a VM). They can also be defined at the rack level, where each *rack flow* consists of all packets from a rack to the Internet or vice versa. They can even be defined at the pod level, where a *pod flow* consists of all packets from a pod to the Internet or vice versa. These flows are organized in a three-level hierarchy, where each pod flow contains multiple rack flows, each rack flow consists of multiple VM flows, and each packet belongs to one flow at each level. Measuring the flow spread, which is the number of distinct elements (under measurement) in a flow, at different levels provide information with different granularities for traffic analysis.

Second, instead of the simple metric of packet number, we can measure any *elements* that are defined according to an application's requirements and carried in the packet headers or payload. Use the previous cloud example. The provider may monitor its clients' outbound traffic for suspicious activities, where each VM (rack or pod) flow consists of all packets from the VM (rack or pod) to the Internet. In particular, it may measure the number of distinct destinations in each flow. For instance, if a VM flow contacts too many destination addresses than it normally does, the VM may be used as a bot for scanning. If a rack or pod flow contains too many destination addresses than normal, even though its individual VMs appear to behave within bound, the overall aggressive behavior at the pod level may signal a botnet activity or a worm activity with many VMs in the pod compromised for *stealthy* worm propagation, where each infected host restrains its scanning rate from being too high to avoid detection.

Summarizing the above discussions, the flow model in a typical datacenter is an hierarchical model. Although hierarchical models are usually applied in a number of multi-layer networks this paper explores the model to reduce traffic measurement overheads in spread estimation. To this end, we proposed a new form of traffic measurement, called *hierarchical spread estimation*, which is not studied before. It estimates flows' spreads, where all flows are organized in a hierarchical structure, offering different granularites in traffic measurement at once.

Measuring flow spread at a single level has been studied before with two classes of solutions: One class measures each flow separately, keeping track of all flow labels

and assigning a separate data structure for each flow to encode its elements [11], [12], [13]; the other class is more memory-efficient by encoding the elements from all flows in a single compact data structure without keeping track of flow labels — given a flow label (obtained by other means or of interest to the admin), the compact data structure can estimate the flow spread [14],[15]. Due to its excellent space-time efficiency, this paper will focus on the second class by extending the problem from a flat single level of flows (such as TCP flows traditionally) to a generalized multi-level hierarchy of flows. One may argue that the traditional single-level solutions can be simply applied to every level of a hierarchy. The problem is that each arrival packet will need to be processed multiple times, one at every level. [16] propose a solution that uses counter to track sizes (number of packets) of hierarchical flows. However, this method is unsuitable for spread estimation because counters cannot keep track of addresses in order to remove duplicates.

The contribution of this paper is to conduct the first study on hierarchical spread estimation, with an efficient solution that physically processes each arrival packet only once, while logically encoding the element of the packet at all levels for all relevant flows. It achieves the benefits of multi-grained traffic measurement at the same cost of traditional single-level solutions. Technically, we propose a new hierarchical virtual bitmap architecture, which shares bits not only among flows at the same level to save cache memory, but also among flows across levels such that encoding an element from an arrival packet at the lowest level of the hierarchy will automatically propagate the encoded information through all levels, regardless of the number of levels there are. We mathematically derive the formula of our hierarchical spread estimator, and prove that the proposed estimator is asymptotically unbiased. We implement the estimator both in software and hardware. Through extensive experiments, we demonstrate that compared to the state-of-the-art, hierarchical virtual bitmap estimator (HVE) delivers from 43% to 155% more throughput while its accuracy is in general close and even has up to 97% smaller error for some large upper-level flows. The rest of the paper is organized as follows. Section 2 presents the flow model, system model and formulates our research problem. Section 3 discusses the related works. We present the detailed architecture of our solution (HVE) in Section 4. Section 5 extensively evaluates HVE. Section 6 concludes the paper.

2 Flow Model, System Model and Problem Statement

2.1 Flow Model

Consider a hierarchical flow model of l levels. Flows at each level are disjoint, while a flow f_{j-1} at the $(j-1)$th level contains multiple child flows f_jth at the jth level, forming a hierarchical structure among flows of different levels, as illustrated in Fig. 1, where $1 < j \leq l$.

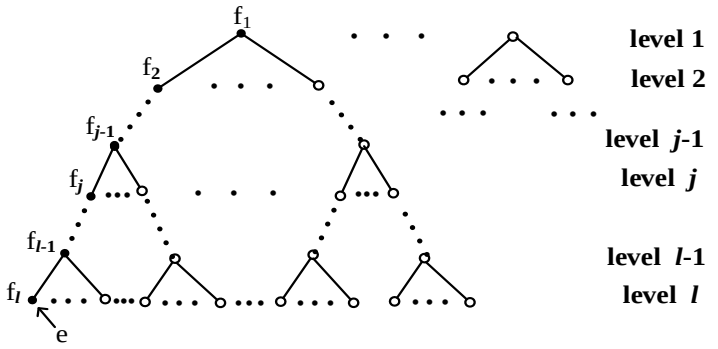


Fig. 1: An example of a hierarchical flow model of l levels.

Each packet belongs to a single flow f_l at the lth level. It also belongs to the flow parent at the $(l-1)$th level, to that flow parent at the $(l-2)$th level, ..., and all the way to a flow at the first level f_1, which contains f_l as a descendant in the hierarchy. Flows at the lth level are called *base flows*; flows at other levels are called *aggregate flows*.

2.2 System Model

Our system consists of gateway routers/switches responsible for traffic measurement. When a packet arrives at a switch/router, the labels of the base/aggregate flows that it belongs to (such as $f_1, \ldots, f_l$) are extracted from the packet header. For example, suppose the source address is the base flow label at the first level, the address 24-bit prefix is the second-level flow label, and the address 16-bit prefix is the third-level flow label. The switch can easily obtain all flow labels by extracting the source address from the packet header and applying appropriate bit masks. After obtaining f_1 through f_l, the switch also extracts the element e, which are defined based on application need and can be found in packet headers or payload from the packet. In case that the destination address is the element, the switch will copy it from the header.

2.3 Problem Statement

Given an arrival packet stream at a switch/router, the problem is to measure the spreads of all flows in the hierarchy, including both base flows and aggregate flows. Our goal is to design a hierarchical spread estimator that encodes each packet only once overall, instead of once for each of l base/aggregate flows it belongs to, yet being able to provide accurate spread estimation for all flows. The design of such an estimator includes two operations:

– Online element encoding: It stores distinct elements from all flows in a compact data structure for online operation at the same place where packet forwarding is performed
– Offline Spread estimation: It takes the encoded data and calculates an estimation for the spread of any given flow.

3 Preliminaries

In this section, we briefly review related methods of estimating a flow spread. Suppose that an incoming flow at a switch w is represented as $\langle f, e\rangle$, where f is a flow id and e is an element id. The spread of f is the number of distinct elements encoded during a measurement period. Let n be the actual spread of f. In what follows, we will discuss how n can be estimated through the methods mentioned above.

3.1 Bitmaps

[11] propose the bitmap as a lightweight and compact data structure to estimate the spread of a flow. In order to estimate n with a bitmap, an array B, which contains m bits initialized to zeroes, is allocated to store distinct elements of f. For each element e, we randomly set a location k^* in B as 1, that is

$$B[k^*] = 1 \tag{1}$$

where $k^* = H(e) \bmod m$ and $H(\cdot)$ is a hash function. Once all contacts are stored in B, n is estimated as

$$\hat{n} = -m \ln V_m \tag{2}$$

where V_m is the fraction of bits in B that are still '0' at the end of the measurement period.

3.2 Opensketch(Bitmap)

A bitmap is ideal for estimating the spread of one flow. If we have to estimate the spreads of multiple flows, then we will need to construct an independent bitmap for each flow. Consequently, the combined size of independent bitmaps is proportional to the number of flows monitored in a measurement period. More compact and memory efficient spread estimators are based on an idea of memory sharing, where all flows' elements are encoded in a shared physical memory.

A memory sharing spread estimator based on CountMin [17] is proposed by OpenSketch [3], which replaces counters in CountMin with uniform bitmaps. We will refer to this estimator as *opensketch(bitmap)*. Let B be an array of k bitmaps.

For each data item $\langle f, e\rangle$, it hashes f to k bitmaps, then hashes e to a bit in each bitmap, and sets that bit to one,

$$B[H_i(f)][H(e)] = 1, 0 \leq i < k. \tag{3}$$

Each of the k bitmaps for f produces an estimate for the flow spread, which carries noise from other flows due to hash collision. Final estimate for the spread of f is the minimum estimate from the k bitmaps, since it contains the least noise.

3.3 Virtual Bitmap

Instead of constructing independent bitmaps, [14] randomly construct *virtual bitmaps* from a shared pool of physical array of bits. Let P be a shared array of m bits. Let the size of each virtual bitmap be s. The virtual bitmap of f, denoted as X_f, is generated in the following way:

$$X_f[i] = P[H_i(f)],\ 0 \leq i < s, \tag{4}$$

where H_i, $0 \leq i < s$, are independent hash functions. For each arrival element e of f, we randomly select a location k^* in X_f as

$$k^* = H(e) \bmod s, \tag{5}$$

where $H(\cdot)$ is a hash function. Next, we set the bit at location k^* in X_f to 1:

$$X_f[k^*] = P[H_{k^*}(f)] = 1. \tag{6}$$

This implies that, through virtual bitmaps, all distinct elements belonging to f and all others flows are stored in P. Unfortunately, X_f not only contains the spread of f, but potentially contains some noise from other flows. Hence, the estimated value of n is

$$\hat{n} = s\ln(V_m) - s\ln(V_s), \tag{7}$$

where V_s and V_m are the fractions of bits that are still '0' in X_f and P, respectively, at the end of a measurement period. The first term on the right hand side of equation (7) is the estimated noise contained in X_f.

3.4 Other Related Works

Our focus in this paper is on estimation of flow spread. Past solutions use hash-based compact data structures called *sketches* [11], [12], [13], [18], [14], [15], [19], [20], [21], [3], [22], [23], [24]. These solutions can be divided into three categories. The first category compactly estimates the spread of a single flow [11], [12], [13], [18], [25]. As a result, the total memory allocation is proportional to the number

of flows being monitored. The second category estimates the spreads of multiple flows by using a shared pool of resources, either bits or counters [14], [15], [19], [20], [21]. This is done by constructing virtual sketches from the shared memory. The third category introduces universal and adaptive sketches. This category not only estimates flow spread, but can also be used to perform other tasks such as identification of heavy hitters and detection of traffic changes [3], [22], [23], [24], [25].

4 Hierarchical Virtual Bitmap Estimator (HVE)

There are two main operations in our Hierarchical Virtual bitmap Estimator (HVE). The first operation deals with online data encoding. The second operation is spread estimation, which is carried out either by the control plane of the switch/router that performs online encoding or by a centralized controller.

4.1 Virtual Arrays

The data structure for HVE is simply an array B of m bits, which are initialized to zeros at the beginning of each measurement period. A bit in the array is denoted as $B[k]$, $0 \leq k < m$. Our approach is to pseudo-randomly allocate a virtual array of bits from B to each base/aggregate flow to encode its elements.

Without lose of generality, consider an arbitrary packet, carrying an element e and belonging to a base flow f_l, whose parent chain is f_1, ..., f_{l-1}. Fig. 2 illustrates how bits are allocated for the virtual arrays of f_l through f_1, which are denoted as B_{f_j}, $1 \leq j \leq l$. Because flows are dependent, we do not independently assign physical bits to flows. Rather, each flow takes some bits from its parent's virtual array to form its own virtual array. Since first-level flows have no parents, they pseudo-randomly select bits from the physical array B.

Let $s_j, 1 \leq j \leq l$ be the pre-defined length for the virtual array of any flow at the jth level, where $s_j < s_{j-1}$, $1 < j \leq l$. We first describe how an arbitrary base flow f_l will select s_l bits for its virtual array, and then describe how an arbitrary jth-level aggregate flow f_j will select s_j bits from its parent f_{j-1}'s virtual array, $1 < j \leq l$. The bits in B_{f_1} are pseudo-randomly selected from B as follows in equation 8.

$$B_{f_1}[k] = B[H_k(f_1)],\ 0 \leq k < s_1, \tag{8}$$

where H_k, $0 \leq k < s_1$, is an independent hash function. We can replace the s_1 hash functions in (8) with a single master hash function H_M as follows:

$$H_k(f_1) = H_M(f_1 \oplus R_1[k]),\ 0 \leq k < s_1, \tag{9}$$

where R_1 is a set of s_1 random numbers and $\oplus$ is the XOR operator. By substituting (9) into (8), we have

$$B_{f_1}[k] = B[H_M(f_1 \oplus R_1[k])],\ 0 \leq k < s_1. \tag{10}$$

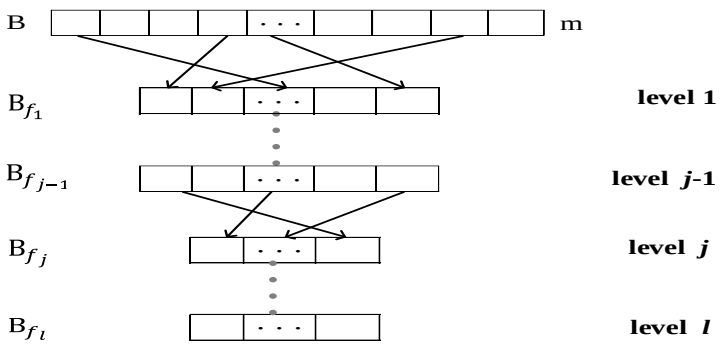


Fig. 2: Bits allocation in hierarchical virtual bitmaps.

Generalizing in equation (11), the bits in B_{f_j} are pseudo-randomly selected from $B_{f_{j-1}}$, where B_{f_j} is the virtual array of flow f_j at the jth level and $B_{f_{j-1}}$ is the virtual array of the parent flow f_{j-1} at the $(j-1)$th level.

$$B_{f_j}[k] = B_{f_{j-1}}[H_M(f_j \oplus R_j[k])],\ 0 \leq k < s_j. \tag{11}$$

These are virtual constructions that are not actually carried out online.

4.2 Online Encoding

A switch that receives an arrival packet, which belongs to f_1,...,f_l, will pseudo-randomly select a bit from the virtual array B_{f_l} of the highest-level flow f_l and encode the element by setting the bit to one. Recall that this bit is taken from its parent's virtual array $B_{f_{l-1}}$. Hence, by setting the bit, we have also encoded the element for the parent flow. As this argument repeats, by setting just one bit, we have actually encoded the elements for all flows f_l through f_1 in their virtual arrays.

However, we cannot operate directly on the virtual arrays, which are virtual after all. The bit we are setting is a physical bit, which is taken in the virtual arrays. Below we show how to select a bit from B_{f_l} to set and how this bit is translated into a physical bit in B for setting. In equation (12), the selection of a bit is done by hashing the element e for an index.

$$k^* = H_M(e) \bmod s_l \tag{12}$$

From equation (11), the virtual bit of B_{f_l} at index k^* is the following bit in at the $(l-1)$th level as in equation (13):

$$B_{f_l}[k^*] = B_{f_{l-1}}[H_M(f_l \oplus R_l[k^*])], \tag{13}$$

which is in turn the following bit at the $(l-2)$th level as in equation (14):

$$B_{f_{l-1}}[H_M(f_l \oplus R_l[k^*])] = B_{f_{l-2}}[H_M(f_{l-1} \oplus R_{l-1}[H_M(f_l \oplus R_l[k^*])])]. \quad (14)$$

Repeating the above process, we eventually reach a bit in the physical array B as in equation (15):

$$B_{f_l}[k^*] = B[H_M(f_1 \oplus R_1[H_M(f_2 \oplus R_2[\cdots H_M(f_l \oplus R_l[k^*])])])] \quad (15)$$

The only encoding action taken by the switch after receiving a packet is to set the above physical bit to one, as done in equation (16),

$$B[H_M(f_1 \oplus R_1[H_M(f_2 \oplus R_2[\cdots H_M(f_l \oplus R_l[k^*])])])] = 1 \quad (16)$$

This assignment automatically encodes the element in all l virtual arrays, B_{f_1}, B_{f_2}, $\cdots B_{f_l}$, for the flows that the packet belongs to, with $l+1$ hashes and one memory access. These hash computations can be pipelined in hardware implementation [16], which is very efficient as we observe in our GPU implementation. The full pipeline implementation encodes each packet in one clock cycle.

4.3 Spread Estimation

At the end of each measurement period, the physical array B is offloaded to the switch's control plane or to a centralized controller, where spread estimation is performed. Given an arbitrary flow label f_j at an arbitrary level $1 \leq j \leq l$, we first derive the labels on its parent chain, f_{l-1} through f_1. We then construct its virtual array B_{f_j} by copying its s_j bits from the physical array B as in equation (17):

$$B[H_M(f_1 \oplus R_1[H_M(f_2 \oplus R_2[\cdots H_M(f_j \oplus R_j[k])])])], 0 \leq k < s_j, \quad (17)$$

where f_1 is the parent flow to f_2, which is in turn the parent flow to f_3, and all the way to f_j. We stress that this construction happens during offline spread estimation, whereas no virtual array is constructed during the online operation of element encoding. We similarly construct the virtual arrays of $B_{f_{j-1}}$ through B_{f_1}. Our proposed spread estimator, HVE, is derived in Theorem 1:

Theorem 1. *Let n be the total number of distinct elements from all flows and n_{f_i} be the actual spread of flow f_i, where $1 \leq i \leq j$. Let U_m be the number of '0' bits in B and U_{f_i} be the number of '0' bits in the virtual array B_{f_i}. We define V_{f_i} as*

$$V_{f_0} = \frac{U_m}{m} \text{ and } V_{f_i} = \frac{U_{f_i}}{s_i}, 1 \leq i \leq j. \quad (18)$$

Then HVE *for n_{f_i} is*

$$\hat{n}_{f_j} \simeq s_j \ln(V_{f_{j-1}}) - s_j \ln(V_{f_j}), \quad (19)$$

when $1 < j \leq l$ *and*

$$\hat{n}_{f_1} \simeq s_1 \ln(V_{f_0}) - s_1 \ln(V_{f_1}), \tag{20}$$

when $j = 1$.

Proof. Before we derive our estimator, we first estimate the $\ln(E(V_{f_j}))$ in equation (21):

$$\ln(E(V_{f_j})) = -\frac{n}{m} - \frac{n_{f_1}}{s_1} - \frac{n_{f_2}}{s_2} - \cdots - \frac{n_{f_j}}{s_j}. \tag{21}$$

Let $A(f_j, k)$ be an event that the bit at an arbitrary index k in the virtual array B_{f_j} remains '0' at the end of a measurement period. Let $I(f_j, k)$ be a binary indicator, which is 1 if $A(f_j, k)$ happens, or 0 otherwise. Let $\hat{k}$ be the index of the physical bit in B that is selected for the bit at index k in B_{f_j}. Event $A(f_j, k)$ occurs *if, and only if,* none of the arrival packets sets the bit at index k in B to 1.

Let b be the bit at index k in B_{f_j}. Each of the n_{f_j} elements from flow f_j has a probability of $\frac{1}{s_j}$ to set the bit b as 1; each of the $n_{f_{j-1}} - n_{f_j}$ elements in its parent flow but not in f_j has a probability of $\frac{1}{s_{j-1}}$ to set the bit b as 1.

Continuing this line of reasoning, each of the $n - n_{f_1}$ elements not in f_1 has a probability of $\frac{1}{m}$ to set the bit b in B. Hence, the probability for $A(f_j, k)$ to happen is stated in equation (22):

$$Prob(A(f_j, k)) = \left(1 - \frac{1}{m}\right)^{n - n_{f_1}} \left(1 - \frac{1}{s_1}\right)^{n_{f_1} - n_{f_2}} \cdots \left(1 - \frac{1}{s_j}\right)^{n_{f_j}}, 0 \leq k \leq s_j - 1. \tag{22}$$

By definition, $U_{f_j} = \sum_{k=0}^{s_j - 1} I(f_j, k)$. Therefore,

$$\begin{aligned} E(V_{f_j}) &= \frac{1}{s_j} \sum_{k=0}^{s_j - 1} E(I(f_j, k)) \\ &= \frac{1}{s_j} \sum_{k=0}^{s_j - 1} Prob(A(f_j, k)) \\ &= \left(1 - \frac{1}{m}\right)^{n - n_{f_1}} \cdots \left(1 - \frac{1}{s_j}\right)^{n_{f_j}} \end{aligned} \tag{23}$$

$E(V_{f_j})$ in (23) can be approximated as in equation (24)

$$E(V_{f_j}) \simeq e^{-\frac{n - n_{f_1}}{m}} e^{-\frac{n_{f_1} - n_{f_2}}{s_1}} \cdots e^{-\frac{n_{f_{j-1}} - n_{f_j}}{s_{j-1}}} e^{-\frac{n_{f_j}}{s_j}}, \tag{24}$$

when m, s_1, ..., s_j, $n - n_{f_1}$, $n_{f_1} - n_{f_2}$, ..., n_{f_j} are sufficiently large. Assume the spread of a child flow is much smaller than the spread of a parent flow (which contains many child flows), i.e., $n_{f_1} \ll n$, $n_{f_2} \ll n_1$, ..., $n_{f_j} \ll n_{j-1}$. We have

$$E(V_{f_j}) \simeq e^{-\frac{n}{m} - \frac{n_{f_1}}{s_1} \cdots - \frac{n_{f_{j-1}}}{s_{j-1}} - \frac{n_{f_j}}{s_j}}. \tag{25}$$

We rewrite equation (25) as equation (26)

$$\ln(E(V_{f_j})) \simeq -\frac{n}{m} - \frac{n_{f_1}}{s_1} \cdots - \frac{n_{f_{j-1}}}{s_{j-1}} - \frac{n_{f_j}}{s_j}. \tag{26}$$

Because (26) holds for any $j \in (1, l]$, it holds for $j - 1$ as well, which means that $\ln(E(V_{f_{j-1}}))$ is approximated in equation (27):

$$\ln(E(V_{f_{j-1}})) \simeq -\frac{n}{m} - \frac{n_{f_1}}{s_1} \cdots - \frac{n_{f_{j-2}}}{s_{j-2}} - \frac{n_{f_{j-1}}}{s_{j-1}}. \tag{27}$$

Combining (26) and (27), the approximate value of $\ln(E(V_{f_j}))$ s given in equation (28):

$$\ln(E(V_{f_j})) \simeq \ln(E(V_{f_{j-1}})) - \frac{n_{f_j}}{s_j}. \tag{28}$$

From (28), n_{f_j} is solved in equation (29):

$$n_{f_j} \simeq s_j \ln(E(V_{f_{j-1}})) - s_j \ln(E(V_{f_j})). \tag{29}$$

By replacing $E(V_{f_j})$ and $E(V_{f_{j-1}})$ with the instance values of V_{f_j} and $V_{f_{j-1}}$ that are directly obtained from the constructed virtual arrays, B_{f_j} and $B_{f_{j-1}}$, respectively, our hierarchical virtual bitmap estimator (HVE) is given in equation (30):

$$\hat{n}_{f_j} \simeq s_j \ln(V_{f_{j-1}}) - s_j \ln(V_{f_j}), \tag{30}$$

where $\hat{n}_{f_j}$ refers to the estimate of true spread n_{f_j}. When $j = 1$, we have the spread of f_1 given in equation (31):

$$\hat{n}_{f_1} \simeq s_1 \ln(V_{f_0}) - s_1 \ln(V_{f_1}), \tag{31}$$

which is consistent with the non-hierarchical estimator of (7).

5 Performance Evaluation

In this section we evaluate the performance of the proposed Hierarchical Virtual bitmap Estimator (HVE) in comparism with prior art, through simulations on CPU and GPU implements.

We use real Internet traces from CAIDA [26] to simulate a two-level hierarchical traffic model. A second-level (base) flow consist of all packets to a destination address in downloaded traces. The spread of base flows are from the range $[1, 5000]$. Aggregate flows are identified by 16-bit prefix of the destination addresses. This classification results in about 4000 aggregate flows whose spreads are in a range $[1, 15000]$.

The most related and state-of-art compact spread estimators that we compare our work with are the virtual bitmaps (VB) [14] and opensketch(bitmap) [3] (check Section 3 for brief descriptions), which are flat single-level spread estimators. All the estimators [HVE, VB and opensketch(bitmap)] attempt to estimate a flow spread in a tight memory but their design goals are somewhat different. The main goal of the VB and opensketch(bitmap) is compact and accurate spread estimation for all types of flows, whether independent or otherwise, through sharing of memory among all flows. On the other hand, HVE tries to increase throughput (which is the number of packets processed per seconds) of hierarchical flows, while at the same time delivering compact spread estimations whose accuracy is comparable to the state-of-art. We implement VB and opsketch(bitmap) in two ways. All aggregate and base flows share the whole available physical memory in the first implementation. However, in the second implementation, one half of the physical memory is allocated to encode aggregate flows, while the other half is allocated to encode base flows. We denote the second implementation as VB* and opensketch(bitmap)*.

We will first compare the throughput of HVE with VB and opensketch(bitmap) for the setup above on CPU and GPU implementations. Afterward, we will compare the accuracy of the three estimators. Our goal is to examine whether our multi-level etimator delivers a superior throughput at the same costs of single-level estimators, that is if HVE is at least as accurate as VB and opensketch(bitmap).

Throughput of HVE: We compare the throughput of HVE with VB and opensketch(bitmap) on CPU and GPU implementations, whose specifications are the following:
CPU Implementation: We implement all the three estimators in Java, version 9, on a multi-core CPU, running Intel(R) Xeon processor @ 3.7GHz. The machine has 32GB of RAM with 2TB HDD and 512 SSD.
GPU Implementation: We use CUDA 10 toolkit to program an NVIDIA GTX 1070 GPU, with 8GB GDDR5 memory @ 1506MHz. The GPU has 1920 cuda cores.
We utilize parallel processing to speed up the online encoding on GPU implementation.

Throughput indicates the processing speed of the three estimators. We compare throughput in Table 1. Clearly, on CPU, HVE processes up to 155% more packets than VB and opensketch(bitmap). Table 1 also shows that HVE processes at least

43% more packets than VB and opensketch(bitmap) on GPU.

Table 1: The throughput (in million packets per second) of HVE, opensketch(bitmap) and VB.

Estimatord	**Software (Mpkt/sec)**	**GPU (Mpkt/sec)**
HVE	4.44	696
opensketch(bitmap)	1.74	445
opensketch(bitmap)*	1.94	423
VB	2.65	487
VB*	2.66	462

Accuracy of HVE: In the previous section, we show that HVE significantly improve throughput compared to state-of-the-art. Here, we will show that this advantage does not degrade its accuracy. In fact, we show that HVE's accuracy is up to that of VB and opensketch(bitmap) and even better for some flows.

We evaluate the accuracy of the three estimators when we allocate $m = 0.25$MB, 1MB, 2MB and 4MB. In our implementation of HVE, we set the lengths of virtual arrays for aggregate and base flows as $s_{f_1} = 64000$ bits and $s_{f_2} = 8000$ bits, respectively. These parameters are chosen to ensure that the average relative error of estimated spread of the flow in the data set above under HVE is less than 5%. However, we obtain a large average relative error when the same parameters are used under virtual bitmap and opensketch(bitmap). This because the larger a virtual bitmap is the more error it accumulates. On the other hand, we calculate the length of (virtual) bitmaps sufficient to estimate the largest spread. Consequently, the length of virtual bitmaps used to estimate the spread of aggregate and base flows under VB and VB* are 3000 bits and 1000 bits, respective. Also, the length of bitmaps used in opensketch(bitmap) and opensketch(bitmap)* is 3000 bits. In our evaluation, the spread of a large flow is at least 1000, while the spread of a small (to medium) flow is less than 1000.

We present our estimation results in Fig. 3 and 4 for aggregate and base flows, respectively, when m =2M. In Plots (a) - (e) of each figure, $x-$axis represents the actual spread of a flow and $y-$axis represents the corresponding estimated spread value. Each point on the plots represents a flow and line $y = x$ is shown for reference, such that the closer a point is to the line, the more accurate the estimation represented by the point. Plot (f) in the figures compares standard error of HVE to that of VB, VB*, opensketch(bitmap) and opensketch(bitmap)*. In the plot, the $x-$axis represents the actual flow spread while the $y-$axis represents standard error of the estimations

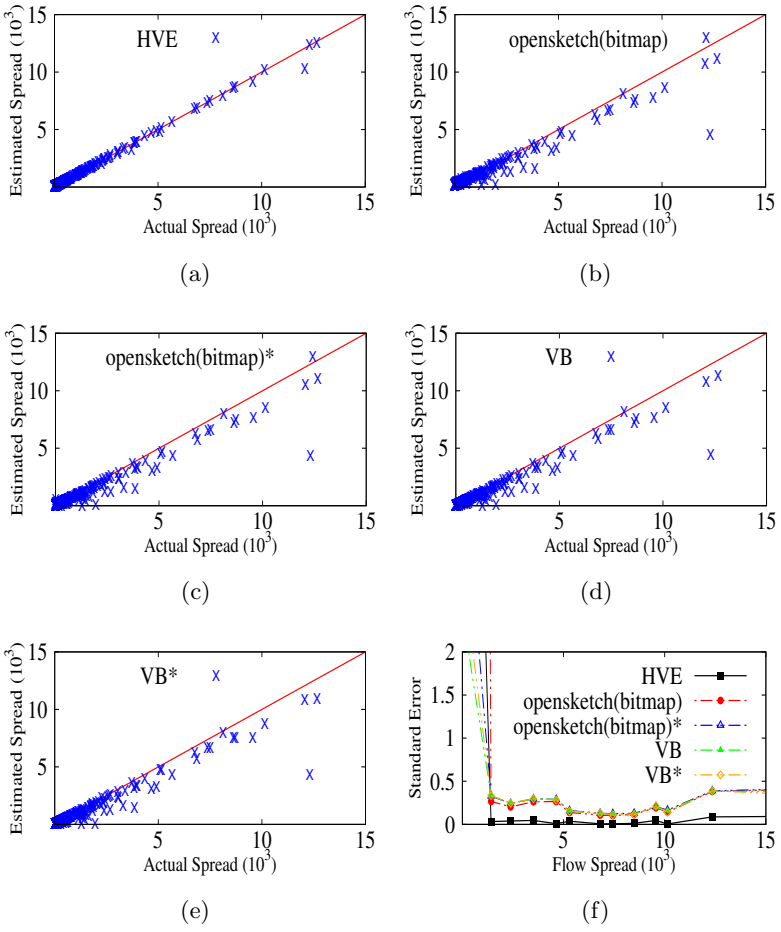


Fig. 3: Accuracy of HVE (plot (a)) vs. opensketch(bitmap) (plots (b) and (c)) vs. VB (plots (d) and (e)) for **aggregate flows**

Plot (a) in Fig. 3 and 4 shows that HVE accurately estimate the spread of aggregate and base flows, respectively, since the points cluster more closely around the equality line. For aggregate flows, Fig. 3 compares (a) HVE with (b) opensketch(bitmap), (c) opensketch(bitmap)*, (d) VB, and (e) VB* and their relative errors in Fig. 3(f). We see that HVE is significantly more accurate for large aggregate flows. We further confirm these results in Table 2, which compares the accuracy of the estimators for aggregate flows when $m = 0.25$MB, 1MB, 2MB and 4MB. The average relative error of large aggregate flows under HVE is at least 77% smaller than other estimators.

Fig. 4 compares the accuracy of base flows under (a) HVE with (b) opensketch(bitmap), (c) opensketch(bitmap)*, (d) VB, and (e) VB* and their relative er-

Table 2: Comparison of average relative error of estimated spreads of large aggregate flows (whose spreads are greater than 1000) under HVE, VB and opensketch(bitmap). Additionally, we also compare the average absolute error of estimated spreads of small aggregate flows (whose spread is less than 1000) under the estimations.

Estimators	**Relative error of large flows (spread $\geq$ 1000)**			**Absolute error of small flows (spread $<$ 1000)**		
	$m = 0.25$MB	$m = 1$MB	$m = 4$MB	$m = 0.25$MB	$m = 1$MB	$m = 4$MB
HVE	0.048	0.029	0.022	61.45	31.62	16.49
opensketch(bitmap)	0.766	0.171	0.156	1875.09	361.64	66.14
opensketch(bitmap)*	0.312	0.214	0.212	744.99	43.72	11.88
VB	0.220	0.219	0.217	25.95	17.66	13.80
VB*	0.217	0.215	0.214	22.30	16.55	13.91

rors in Fig. 4(f). Clearly, the accuracy of HVE is up to VB and opensketch(bitmap). In particular, as shown in Table 3, which compares the accuracy of the estimators for base flows when $m = 0.25$MB, 1MB, 2MB and 4MB, the average relative error of HVE for large base flows is at least 63% and up to 97% smaller than opensketch(bitmap), while its accuracy is similar to VB (in the worst case).

Although the average absolute error for small base and aggregate flows is larger for HVE compared with VB and VB*, it is still very small relative to actual spreads (about 1.7% on average, when $m = 4$MB). Note that for large aggregate and base flows, the accuracy of HVE is up to (or surpass in certain cases) the other estimators. This is important because accurate estimation of large spreads is needed for essential network management tasks, such as superspreader identification.

Table 3: Comparison of average relative error of estimated spreads of large base flows (whose spread is greater than 1000) under the HVE, VB and opensketch(bitmap). Additionally, we also compare the average absolute error of estimated spreads of small base flows (whose spread is less than 1000) the estimators.

Estimators	**Relative error of large flows (spread $\geq$ 1000)**			**Absolute error of small flows (spread $<$ 1000)**		
	$m = 0.25$MB	$m = 1$MB	$m = 4$MB	$m = 0.25$MB	$m = 1$MB	$m = 4$MB
HVE	0.021	0.018	0.013	27.41	19.77	16.59
opensketch(bitmap)	0.974	0.197	0.036	1883.84	370.48	70.031
opensketch(bitmap)*	0.431	0.142	0.081	939.26	204.91	39.74
VB	0.027	0.017	0.015	17.97	9.97	5.29
VB*	0.038	0.032	0.031	13.43	6.87	3.67

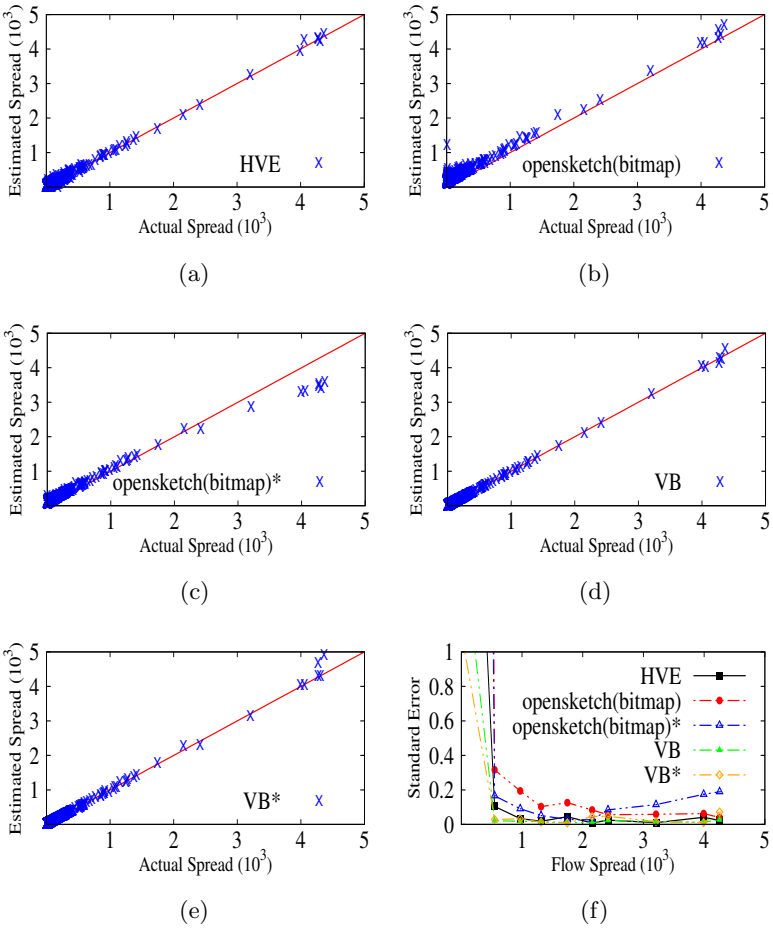


Fig. 4: Accuracy of HVE (plot (a)) vs. opensketch(bitmap) (plots (b) and (c)) vs. VB (plots (d) and (e)) for **base flows**

6 Conclusions

This paper proposes a new hierarchical measurement architecture by introducing hierarchical virtual bitmaps estimator, which extends the capability of existing single-level network traffic measurement tools and enables more efficient online data encoding of flows with hierarchical structure. We mathematically derive a hierarchical spread estimator that enables multi-level spread estimation at the same costs of single-level spread estimators. Finally, through CPU and GPU implementations, we show that our hierarchical virtual bitmap estimator's throughput significantly exceeds prior art while its accuracy is in general comparable with (or at times

better than that of) prior art. The future works include rigorus analysis of HVE's accuracy, that is proving HVE unbiasness and deriving its confidence interval.

7 Acknowledgment

This work was supported by National Science Foundation of US under grants CNS-1909077 and CNS-1719222.

References

1. M. Moshref, M. Yu, and R. Govindan, "Resource/accuracy tradeoffs in software-defined measurement," in *Proceedings of HotSDN*. IEEE, 2015.
2. T. Li, S. Chen, and Y. Ling, "Fast and compact per-flow traffic measurement through randomized counter sharing," in *Proceedings of INFOCOM*. IEEE, 2011.
3. M. Yu, L. Jose, and R. Miao, "Software defined traffic measurement with opensketch," in *Symposium on Networked Systems Design and Implementation*. USENIX, 2013.
4. V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen, "Csamp: A system for network-wide flow monitoring," in *Symposium on Networked Systems Design and Implementation*. USENIX, 2008.
5. Y. Yu, C. Qian, and X. Li, "Distributed and collaborative traffic monitoring in software defined networks," in *Proceedings of HotSDN*. IEEE, 2014.
6. Y. Du, H. Huang, Y. e Sun, S. Chen, and G. Gao, "Self-adaptive sampling for network traffic measurement," in *in Proceeding of IEEE INFOCOM*, 2021.
7. O. Odegbile, S. Chen, D. Melissourgos, and Y. Wang, "Accurate hierarchical traffic measurement in datacenters through differentiated memory allocation," in *Proceedings of the 6th International Conference on Big Data Computing and Communications (BigCom)*, 2020.
8. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 15, Jul. 2009.
9. R. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attack detection using nox/openflow," in *Conference on Local Computer Networks*. IEEE, 2010.
10. Netflow configuration guide. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/15-mt/nf-15-mt-book.html
11. K.-Y. Whang, B. T. Vander-Zanden, and H. M. Taylor, "A linear-time probabilistic counting algorithm for database applications," *in ACM Transactions on Database Systems*, vol. 15, no. 2, pp. 208–229, Jun. 1990.
12. P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier, "Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm," in *Proceedings of Conference on Analysis of Algorithm*, 2007.
13. D. M. Kane, J. Nelson, and D. P. Woodruff, "An optimal algorithm for the distinct elements problem," in *Symposium on Principles of database systems*, 2010, pp. 41–52.
14. M. Yoon, T. Li, S. Chen, and J. Peir, "Fit a compact spread estimator in small high-speed memory," *in IEEE/ACM Transactions on Networking*, vol. 19, no. 5, Oct. 2011.
15. Q. Xiao, S. Chen, M. Chen, and Y. Ling, "Hyper-compact virtual estimators for big network data based on register sharing," in *Proceedings of ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, 2015, pp. 417–428.
16. S. Chen, Y. Zhou, and S. Chen, "Efficient hierarchical traffic measurement in software-defined datacenter networks," in *10th International Conference on Cloud Computing*. IEEE, 2017.
17. G. Cormode and S. Muthukrishnan, "An improved data stream summary: The count-min," *Algorithms*.

18. Q. Xiao, S. Chen, Y. Zhou, and J. Luo, "Estimating cardinality for arbitrarily large data stream with improved memory efficiency," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 433–446, 2020.
19. H. Wang, C. Ma, O. Odegbile, S. Chen, and J.-K. Peir, "Randomized error removal for online spread estimation in data streaming," in *in Proceeding of VLDB Endowment (PVLDB)*, 2021.
20. H. Huang, Y. e Sun, C. Ma, S. Chen, Y. Zhou, W. Yang, S. Tang, H. Xu, and Q. Yan, "An efficient k-persistent spread estimator for traffic measurement in high-speed networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1463–1476, Aug. 2020.
21. Y. e Sun, H. Huang, C. Ma, S. Chen, Y. Du, and Q. Xiao, "Online spread estimation with non-duplicate sampling, proc. of ieee infocom," in *Proceedings of IEEE INFOCOM)*, 2020.
22. T. Yang, J. Jiang, P. Liu, and Q. Huang, "Elastic sketch: adaptive and fast network-wide measurements," in *Conference of Special Interest Group on Data Communication.* ACM, 2018, pp. 561–575.
23. Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with univmon," in *Conference of Special Interest Group on Data Communication.* ACM, 2016, pp. 101–114.
24. Q. Xiao, Z. Tang, and S. Chen, "Universal online sketch for tracking heavy hitters and estimating moments of data streams," in *Proceedings of IEEE INFOCOM)*, 2020.
25. Y. Zhou*, Y. Zhang*, C. Ma, S. Chen, and O. Odegbile, "Generalized sketch families for network traffic measurement," in *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)*, 2019, (* co-first authors).
26. Caida data - overview of datasets, monitors, and reports. [Online]. Available: https://www.caida.org/data/overview/

AUTHOR INDEX