SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

4th SLAAI - International Conference on Artificial Intelligence

(SLAAI-ICAI-2020)

Volume 4

Proceedings of the 16th Annual Sessions
01st December 2020
Via Zoom

Sri Lanka Association for Artificial Intelligence – 2020

SLAAI International Conference on Artificial Intelligence
01st of December 2020
Colombo,
Sri Lanka

The abstracts of all the peer-reviewed and accepted papers that have been presented in the conference are included in this book and the selected full papers are published in a SCOPUS Indexed source.

Editors:

Prof. AS Karunananda
Dr. (Mrs.) ATP Silva
Dr. (Mrs.) WAC Weerakoon
Mr. D Handapangoda

Sri Lanka Association for Artificial Intelligence, Department of Electrical & Computer Engineering, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka.

Contents

Page

-Message from SLAAI President -

It is with great pleasure that I convey this message as the president of Sri Lanka Association for Artificial Intelligence (SLAAI) to the SLAAI- International Conference on Artificial Intelligence 2020 (SLAAI-ICAI2020). SLAAI which was formally formed in the year 2000 has reached to 20 years of unprecedented work in popularization of AI, research in AI and developing Industry- Academia partnerships along with capacity building of the members and those who are interested in AI. During these 20 years journey SLAAI has gradually progressed to what it is today by having annual sessions, local conferences and international conferences that provide an excellent platform for academia, industry and researches to discuss and present their research work and innovations. Tech talks, discussions and meetups are common events in the annual calendar of the SLAAI.

This year marking a new milestone the 4th SLAAI-ICAI conference will be held as a fully online conference due to the prevailing pandemic situation. Even with the numerous difficulties encountered by the mass population due to Covid-19 situation SLAAI has been able to attract quiet a number of papers and the acceptance rate was maintained around 40% after a rigorous double-blind peer review process. I take this opportunity to thank all the authors for their contributions to SLAAI and I would like to extend my heartiest wishes for the authors whose papers have been accepted to be presented at the SLAAI-ICAI 2020.

The conference theme of this year "AI for Global sustainable Development" is a much valid topic in the troubled world today where people are looking for all possibilities of survival in their day-to-day activities as well as conduct in the long term. The richness of AI can be harnessed to address many of the problems faced by the world today and this is being addressed in the seventeen (17) UN Sustainable Development Goals as well.

The conference marks the final event of the year and I would like to thank the keynote speakers, session chairs, panel of reviewers, the program committee, authors and presenters who supported the conference wholeheartedly. I take this opportunity to thank and acknowledge all the hard work and dedication of the members of the SLAAI council 2020 who supported me persistently to raise our standards from physical platform to the online platform enabling the global access more easily.

I wish SLAAI-ICAI-2020 all the very best and to be one of the most productive conference of the year 2020.

Dr. (Mrs.) D.D.M. Ranasinghe
President/ Sri Lanka Association for Artificial Intelligence (SLAAI)
01.12.2020

- Message from Conference Chair -

It is my great pleasure to send this message to 4th SLAAI-International Conference on Artificial Intelligence and the 16th Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI). In 2000, SLAAI was established to promote AI Education and Research in Sri Lanka. SLAAI was founded as the successor to the AI Research group in Sri Lanka (AIRLK). In this year, SLAAI has reached the 21st milestone of its journey by exemplifying its constant contribution to (a) popularize AI among general public, (b) promote AI research and developments, (c) establish industry academia partnership in AI in Sri Lanka. SLAAS has a partnership with SLASSCOM which is the apex body of the Sri Lankan Software industry. Over the last 20 years, SLAAI has popularized AI in numerous ways including, talks at schools, AI meetups - organized jointly with Virtusa (Pvt) Ltd., AI-Asia Summit jointly organized by SLAAI and SLASSCOM has been a forum for popularizing AI, building industry-academia partnerships, and deliberation of research and developments in AI at the international level.

Among other initiatives, SLAAI - International Conference on Artificial Intelligence (SLAAI-ICAI) has been a popular forum for an international community to present and publish their researches covering a wider spectrum of areas of AI including Ontology, Machine Learning, Deep Learning, Natural Language Processing, Machine Translation, Robotics, Bioinformatics, Multi-Agent Systems, Evolutionary Computing, Fuzzy Logic, Theory of computing, and Brain Machine Interfacing. SLAAI promotes theory and practices of both symbolic-AI and non-symbolic-AI. Due to COVID-19 pandemic, although the number of research papers submitted for SLAAI-ICAI-2020 has shown a slight reduction, we manage to select high quality papers with 40% acceptance rate. Research papers submitted to SLAAI go through a rigorous double-blind peer review process and ensure selecting high-quality research papers for the conference. All the papers selected and presented online at SLAAI-ICAI-20 will be published in the SLAAI conference proceedings. All papers from the proceedings undergo through a review process and selected and revised papers will be published in a SCOPUS index source.

Finally, I take this opportunity to express my sincere thanks to the keynote speaker, session chairs, reviewers, panel of judges, authors of the papers, program committee and council members of SLAAI for their untiring effort to make this event a success despite this difficult pandemic situation.

Prof. Asoka S Karunananda
Conference Chair / SLAAI-ICAI-2020
01.12.2020

-Keynote Speech I-

Using Self-Modeling Networks to Model Adaptive Causality

Professor Jan Treur
Social AI Group, Vrije Universiteit of Amsterdam, Netherlands

Abstract. Causal modeling is an intuitive, declarative way of modeling that due to the universal character of causality in principle applies to practically all disciplines. In spite of this seemingly very wide scope of applicability, there are also serious limitations and challenges that stand in the way of applicability, in particular when dynamics and adaptivity play a role. This paper addresses these challenges by exploiting the notion self-modeling network developed from a Network Science perspective. It is shown how temporal-causal networks allow modeling dynamics based on a given network of causal relations and how any causal network can be extended to a self-modeling network to also model dynamic changes of the causal relations themselves. In this way, self-modeling causal networks are obtained that show dynamics of the states of the nodes based on the causal relations as well as adaptivity of these causal relations. Adaptivity is obtained by adding self-models to a given causal base network; these self-models represent the base network's causal structure by additional network nodes for the causal relations that are adaptive. The obtained self-modeling causal networks are themselves temporal-causal networks as well, are still specified in a declarative manner by mathematical relations and functions, and create a next level for the causal network by which the adaptation is addressed. Moreover, this construction can easily be iterated so that multiple orders of adaptation can be covered in the form of multilevel causal models, for example, addressing controlled adaptation or both plasticity and metaplasticity. So, this indeed takes causal modeling to a next level in more than one way so that now dynamics and adaptivity are also covered well, which substantially widens the scope of applicability of causal modeling.

Prof. Jan Treur is a full professor in Artificial Intelligence is an internationally well-recognized expert in human-directed AI and cognitive and social modelling. The research of Prof. Jan Treur during the past 10 years concerns both fundamental and application-directed aspects of human-directed AI. This covers methods and techniques for modelling and analysis of human-directed AI approaches in a number of application areas, including Cognitive and Social modelling and simulation. Prof. Treur is an active author and serves as a PC member in practically all relevant conferences and journals in these AI and application areas. Currently Prof. Treur has focused mostly on multidisciplinary research Network-Oriented Modeling approaches based on adaptive temporal-causal networks to model cognitive, affective and social interactions. Prof. Treur has published two books in the same regard in 2016 and 2020.

More details can be found at URL: https://www.researchgate.net/profile/Jan_Treur

-Keynote Speech II-

Scalable Reinforcement Learning for Engineering Applications

Dr. Varuna De Silva
Institute for Digital Technologies
Loughborough University London

Abstract. Explainability and generalizability are two key demands placed upon any method of Artificial Intelligence for success in the real-world. Furthermore, there is a growing demand for distributed intelligent systems that could coordinate decisions. Reinforcement learning is a branch of Artificial Intelligence with a goal to develop intelligent agents by utilizing trial-and-error techniques. This talk will focus on recent developments in reinforcement learning research that aim to address the challenges of explainability and generalizability by leveraging the advances in neuro-symbolic learning, deep learning and variational inference on probabilistic graphical models. Recent developments in multi-agent reinforcement learning and their engineering applications will also be introduced. Importantly, the talk will discuss various techniques such as curriculum learning, evolutionary methods, and hybrid simulation and data driven methods to scale reinforcement learning research towards application in safety-critical, data-constrained Engineering applications.

Dr. Varuna De Silva is a Senior Lecturer in Machine Intelligence at Loughborough University. He obtained his Ph.D. in video coding and communications from Center for Vision Speech and Signal Processing at University of Surrey in 2011. He has worked in 3 major European Union Funded Projects during 2010-2013. Between 2013 and 2016, Varuna worked as a senior research engineer in computer vision at Apical Ltd (Now part of ARM UK). He currently supervises 7 PhDs and 2 Post-doctoral research fellows in the area of Artificial Intelligence (Multimodal machine learning and multi-agent reinforcement learning) with a strong focus on engineering applications in driverless vehicle technology and team sport analytics.

Deep Learning Approach to Recognition of Novel COVID-19 Using CT Scans and Digital Image Processing

H.M.K.K.M.B. Herath[1] , G.M.K.B. Karunasena [2], S.V.A.S.H. Ariyathunge [3], H.D.N.S. Priyankara [4],
B.G.D.A. Madhusanka[5], H.M.W.T. Herath[6], and U.D.C. Nimanthi[7]
[1,2,3,4,5] Faculty of Engineering Technology, The Open University of Sri Lanka
[6] Faculty of Engineering, University of Moratuwa - Sri Lanka
[7] National Eye Hospital of Sri Lanka, Colombo 10 - Sri Lanka
email: [1]kasunherathlive@gmail.com, [2]kosalabandara91@gmail.com, [3]svasha91@gmail.com, [4]hdpri@ou.ac.lk, [5]bgmad@ou.ac.lk,
[6]windhyaherath@gmail.com

***Abstract*— COVID-19 was announced as a global pandemic by the World Health Organization (WHO) in March 2020. With more than 31.3 million confirmed cases and over 965 thousand deaths recorded as of September 2020, it has inflicted catastrophic damage worldwide. The aim of this study is to development of an algorithm based on artificial intelligence (AI) and image processing techniques to identify COVID-19 patients with the aid of CT chest scan images. This study used a CT scan image dataset that is publically available for the researchers at Kaggle. We were randomly extracted 27% of positive CT (pCT) images and 11% of negative CT (nCT) images from the original dataset. In the testing process, 120 of the test subjects in both nCT and pCT were used to validate the algorithm. Based on the experimental findings, the proposed COVID-19 detection algorithm shows promising results for the identification of COVID-19 patients with 90.83% accuracy at an average precision of 0.905.**

Keywords—Artificial Intelligence (AI), COVID-19 Pneumonia, CT-Scan, Deep Learning, Ground-Glass Opacity (GGO), Machine Vision, SARS-CoV-2)

I. INTRODUCTION

Coronavirus is a large family of viruses that can cause a human being to develop a serious illness. The first reported major epidemic was Severe Acute Respiratory Syndrome (SARS) [1] in 2003, while the second severe outbreak of Middle East Respiratory Syndrome (MERS) [2] in Saudi Arabia began in 2012. The latest outbreak of coronavirus disease was announced in late December 2019. This new virus is very infectious and has spread globally rapidly. On January 30, 2020, as it had spread to 18 countries, the World Health Organization (WHO) declared this outbreak a Public Health Emergency of International Concern (PHEIC) [3]. This virus was named 'COVID-19' by the World Health Organization on February 11, 2020 [4]. As of September 2020, the WHO reported that 31.3 million confirmed cases and over 965 thousand deaths have been registered in 213 countries.

Figure 1 shows confirmed cases of global COVID-19 as of September 2020. The disease has spread rapidly around the globe since it was first identified and has become an international concern. An analysis performed by Jiang et al. [5] found that COVID-19's death rate is 4.5% worldwide. In the age group of 70-79 years, the death rate for patients is 8.0%, while 14.8% for patients over 80 years. Patients over 50 years of age with chronic diseases are at the highest risk and it is critically important to find a way to detect illness before getting into serious conditions.

As the COVID-19 epidemic has become a global pandemic, real-time analysis of epidemiological data is required to prepare society for better disease response plans. COVID-19 belongs to the SARS-CoV and MERS-CoV families, where symptoms of the common cold to severe respiratory diseases, causing trouble breathing, exhaustion, fever, and dry cough, start at the initial level. Real-time Reverse Transcription-Polymerase Chain Reaction or also known as RT-PCR is the latest approach used to make a definitive diagnosis of SARS-CoV-2 infection [6]. PCR testing was found to have a high specificity (Sp) but rather low sensitivity (Sn) with a reported positive rate of only 38%~57%. In addition to etiological laboratory confirmation, Clinical Features (CFs) and chest Computed Tomography (CT) imaging include other key diagnostic elements that could facilitate the identification of COVID-19 pneumonia.

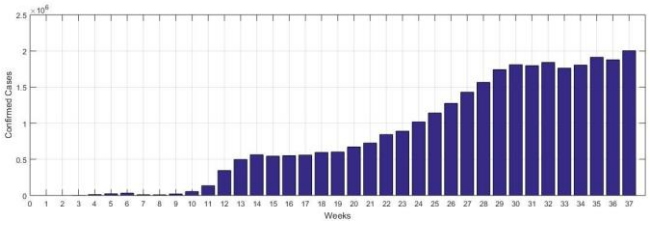


Figure 1 Global COVID-19 confirmed cases as of September 2020

Early identification of patients with COVID-19 pneumonia for timely treatment is crucial to contain the spread, particularly in epidemic regions. According to information shared by the Radiological Society of North America (RSNA), X-ray and CT images of a Chinese person dead by COVID-19 showed the damages done to the human lungs. A research team lead by Lucas [7] at The University of São Paulo demonstrated the chest imaging finding of COVID-19 on different modalities such as Chest Radiography (CXR), Computed Tomography (CT), and Ultrasonography. According to them, chest CT is the main imaging method used in the assessment of COVID-19 pneumonia. A structured chest CT report standardizes imaging results and optimizes contact with the prescribing physician, making it a valuable tool in the pandemic scenario. In addition, the CT imaging properties of infected lungs include Ground-Glass Opacity (GGO) and severity-correlated consolidation. In Hubei Province, China, CT scans have been used widely and on presentation in an attempt to rapidly detect, isolate, and control the spread of the epidemic.

Many studies have documented a high degree of chest CT sensitivity in the diagnosis of COVID-19 pneumonia. Previous studies have shown that the most common CT characteristic of COVID-19 pneumonia is the presence of multifocal Ground-Glass Opacity (GGOs). Figure 2 displays the CT scan image of a COVID-19 patient. Arrowheads reveal the recognizable hazy area on the outer edges of the

lungs. As per the description, Ground-Glass Opacities refer to the distorted presence of the lungs in imaging experiments, almost as though parts were obscured by Ground-Glass. This may be due to the fluid filling of pulmonary airspace, the collapsing of airspace, or both. This is a trend that can be seen while the lungs are sick. Regular lung's CT scans look black; rare chest CTs with GGOs reveal lighter colored or gray spots. Consolidation refers to the saturation of fluids or other inflammatory products in pulmonary airspace. Pleural effusion refers to abnormal fluids that form in the spaces surrounding the lungs.

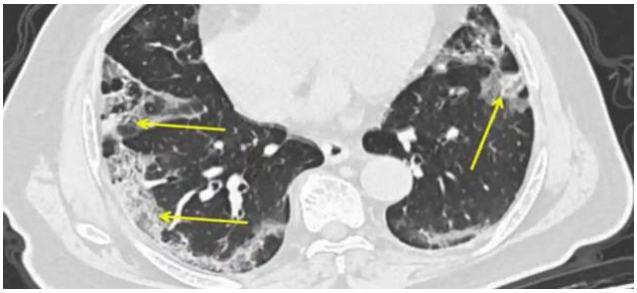


Figure 2 CT scan image of a patient with severe COVID-19, (Photograph: Mount Sinai Hospital).

PCR tests are taking time to diagnose COVID-19 patients and the test results appear to be of low accuracy compared to CT scan tests. However, CT scans can be used as a simple and quick way of categorizing patients into "probably positive" and "probably negative" cohorts. As the hospital admission rate of COVID-19 patients increases, the PCR test is not appropriate. Nowadays, tools for the identification of COVID-19 patients with high efficacy and accuracy are essential. Due to the poor contrast of infection regions of CT images and the large differences in both the shape and location of the lesions in different patients, the delineation of the infection regions in CT scans in the chest is very difficult for the physician. Image processing techniques may open new pathways to describe the state of the lungs using CT scans. The objective of this study is to develop a deep learning algorithm to detect COVID-19 patients using CT chest scan images and validate results for both COVID-19 positive and healthy test subjects.

II. Related Works

Scientific, technological knowledge and resources have been widely used to prevent COVID-19 globally. The number of studies related to the novel COVID-19 is increasing daily. Researchers have recently used imaging patterns on chest CT to detect COVID-19 infection. Research contribution in the field of machine vision and artificial intelligence to prevention novel COVID-19 is described in this section.

Previous studies led by Xu and his research team categorized CT scan images of COVID-19 patients in three groups as healthy cases, Influenza viral pneumonia, and COVID-19 [8]. This study used 175 images of healthy people, 224 images of patients with Influenza-A Pneumonia, and 219 images of patients infected with the coronavirus. The overall accuracy of 86.7% has observed using the 3D-deep learning model.

Shan et al. [9] have developed a method focused on a deep learning mechanism for the segmentation and quantification of contaminated regions and the entire lung using CT images in the chest. A total of 249 COVID-19 patients and 300 new COVID-19 patients have used for validation in their study. They used the Dice similarity 2 coefficient concepts and achieved 91.6% accuracy.

Sachin Sharma [10] from the Institute of Advanced Research of India has engaged with a study about the role of machine learning techniques in obtaining important insights, such as whether a lung CT scan is a first screening/alternative test for RT-PCR. Training and testing have carried out using custom vision software based on Microsoft Azure machine learning techniques. The accuracy of nearly 91% has reached, although some false indicators were found in their analysis.

Harmon [11] and her research team from the USA have shown that a number of deep learning algorithms have been trained in a multi-national cohort of 1,280 patients to locate parietal pleura/lung parenchyma followed by a COVID-19 pneumonia classification. They achieved 90.8% accuracy, with 84% sensitivity and 93% specificity.

Xavier [12] has engaged in a study to evaluate the performance of Artificial Intelligence methods to detect COVID-19 using chest X-Rays and CT scan images. A total of 363 patients have used by combining two different data sources. 191 patients have COVID-19 positive and the rest of them were healthy subjects. The accuracy of the proposed system has reached to 90.9% for the 121 testing samples.

Table 1 Advantages and disadvantages of related works

Ref	Advantages	Disadvantages
[8]	High accuracy achieved (86.7%), The promising results of an additional diagnostic tool for frontline clinical physicians.	The CT manifestation of COVID-19 contrasted only with that of IAVP. In this analysis, A limited number of model samples have used. A limited range of testing and training samples utilized.
[9]	High accuracy (91.6%) achieved large dataset for training and testing has used.	Validation data of CT datasets have collected in one location, which may not be indicative of all COVID-19 patients in other geographical regions.
[10]	CT scan images obtained from various geological sites (Italy, China, Moscow, and India), As the model based on the CT chest images showed strong results in terms of precision and time-consuming.	A high false detection rate has observed in some experiments. A limited range of test samples has used for the study in each geological location.
[11]	High accuracy showed (90.8%), Larger dataset has used for training and testing. The multinational dataset has used to cover the different geological locations of the world.	Model training has limited to patients with positive RT-PCR testing and COVID-19 related pneumonia on chest CT.

[12]	The results have achieved with high accuracy (91.8%), Larger dataset has used for the study.	They also state that it is unknown if the tested procedures may be used to diagnose asymptomatic patients.

Table 1 depicts the advantages and disadvantages of previous related studies referred. From a systematic study, it has been found that CT images of the chest can be used for the early classification of COVID-19 infected patients. Therefore, Convolutional Neural Networks (CNN) model was used in this study to distinguish COVID-19 patient's identification using the CT scan images of the chest.

III. METHODOLOGY

Ground-Glass Opacities (GGO) [13], consolidation, and pleural effusion are the characteristics that are seen as the primary features used in the CT scan picture of a COVID-19 patient. This study mainly focused on detecting the GGO features based on the presence of the COVID-19 infection of the human lungs. The proposed methodology of the system consists of two separate sections, such as the selected CT scan image dataset (section 3.1) and the structure of the COVID-19 artificial intelligence (AI) algorithm (section 3.2). The MATLAB development environment was used to develop the proposed algorithm.

A. CT Image Dataset

This study used the CT scan image dataset from the Kaggle [14] which is publically available for the researchers. The dataset consists of three types of CT images obtained from Union Hospital (HUST-UH) and Liyuan Hospital (HUST-LH). The dataset consists of non-informative CT (NiCT) images, positive CT (pCT) images, and negative CT (nCT) images. We were randomly extracted 27% of pCT and 11% of nCT data for this study. In the testing process, 120 of the extracted data in nCT and pCT were used to validate the method. Table 2 depicts the dataset description used in the study. All the images in the dataset were originally sized to 512×512 pixels.

Table 2 Description of the COVID-19 positive and negative CT scan image dataset

CT Image Type	Data used	Description
Positive CT – pCT	27% of data from original dataset	Imaging features are associated with the COVID-19 Pneumonia
Negative CT – nCT	11% of data from original dataset	Imaging features in both lungs were irrelevant to the COVID-19 Pneumonia

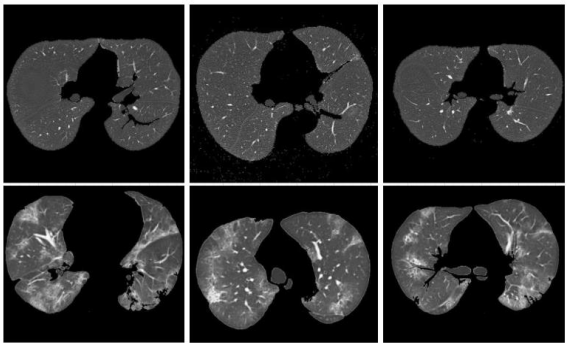


Figure 3 . CT scan images of COVID-19 negative (top row) and positive (bottom row) patients.

In terms of lung changes, the presence of various types of lungs was observed in COVID-19 positive patients. Figure 3 illustrates the CT chest scan images of the COVID 19 positive and healthy test subject's lungs.

B. Proposed CNN Architecture

As the initial phase of this study, the chest CT scan images of COVID-19 subjects and normal healthy subjects are taken and stored in the computer. Then we have performed some image pre-processing steps, such as image cropping and image resizing to extract effective pulmonary regions before using the dataset.

Convolutional Neural Networks (CNN) is a versatile method that is commonly used for image classification. The hierarchical structure and the powerful functionality of image extraction render CNN a complex model for image classification. The proposed CNN architecture composed of two stages: a feature learning stage and a classification stage as shown in Figure 4.

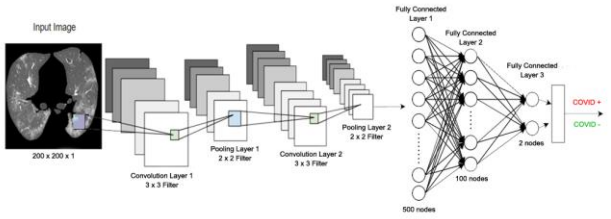


Figure 4 Proposed CNN architecture of the COVID-19 detection algorithm.

The developed feature learning step consists of two convolutional layers and two pooling layers. The first convolution layer includes a 3×3 convolutional filter for initial feature extraction. Then resultant features passed into the first pooling layer which consists of 2×2 max-pooling filters. Then, the extracted features from the first convolution and pooling passed to the second convolution and pooling layers. Furthermore, the second convolution layer consists of 3×3 convolutional filters and the pooling layer consists of 2×2 max-pooling filters. In the classification stage, the feature score matrix passed into a fully connected layer which consists of fully connected three neural layers. Each layer includes 500, 100, and 2 artificial neurons. Finally, the softmaxLayer used to obtain probability and classify input samples to indicate whether the test subjects are COVID

positive or negative. In this model, we have used a 200×200 size input layer.

The Model Hyperparameters are properties that control the whole training process. These include the variables that determine the structure of the network and the variables that determine how the network is trained. The Stochastic Gradient Descent with Momentum (SGDM) optimizer was used as the solver of the training network. The ReLU activation function was used to activate the nodes. The initial learning rate of 0.1 and 0.01 learn rate drop factor was observed at the 20 maximum epochs. We used a mini-batch with 20 observations in each iteration.

IV. Results and Analysis

The proposed design was tested with 120 randomly selected nCT and pCT chest images from the extracted dataset. Figure 5 illustrates the COVID-19 positive test subjects which are lungs have filled with hazy areas. Hazy areas suggested that patients have COVID-19 infection in the body at what level. These subjects are identified as the COVID-19 positive patients by the clinical trials.

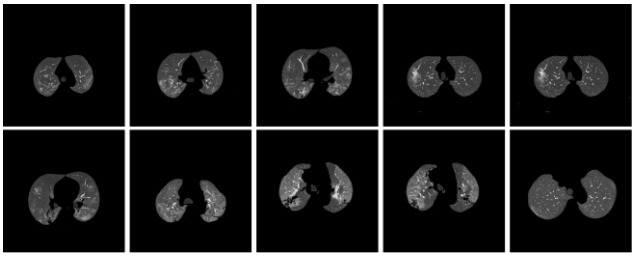


Figure 5 CT chest scan images for COVID-19 positive test subjects.

Figure 6 depicts the sample of healthy test subjects used in this study. According to the images, lungs are observed to be clear and detect less gray spots. Detection of less gray spots suggested that the test subject is negative from COVID-19. These healthy test subjects are identified as COVID-19 negative by the clinical trials.

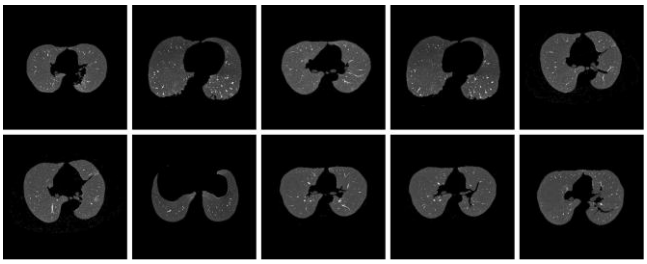


Figure 6 CT chest scan images for healthy test subjects (COVID-19 negative test subjects).

For the training, a classification model was developed. A total of 20 epochs and 2000 iterations were undertaken during the training process in order to achieve optimal model parameters. The accuracy curve of the training process is shown in Figure 7. Based on the training results, the average classification accuracy for each individual mini-batch was 94.25% and the classification accuracy for each individual mini-batch was reached to the maximum at epoch 8.

According to the mini-batch loss curve shown in Figure 8, the mini-batch loss for multi-class classification decreased from 0.695 to 0.0977 at the end of the 20 epochs Based on the test results, positive subjects for COVID-19 were classified with a range of 0.65-1.00 probabilities and healthy subjects (COVID-19 negative subjects) ranged from 0.10-0.40 probabilities. Figure 9 illustrates the experiment results for 120 test subjects of COVID-19 positive and healthy subjects (COVID-19 negative).

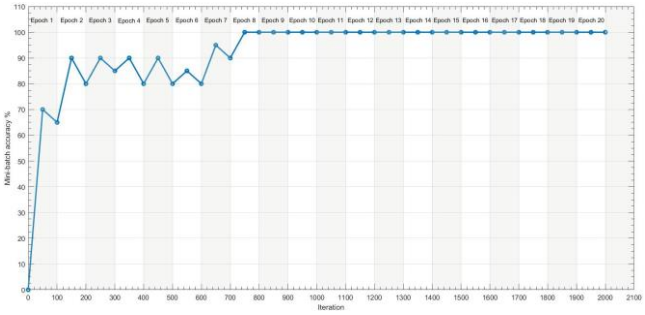


Figure 7 Mini-batch accuracy of the model training for 20 epochs.

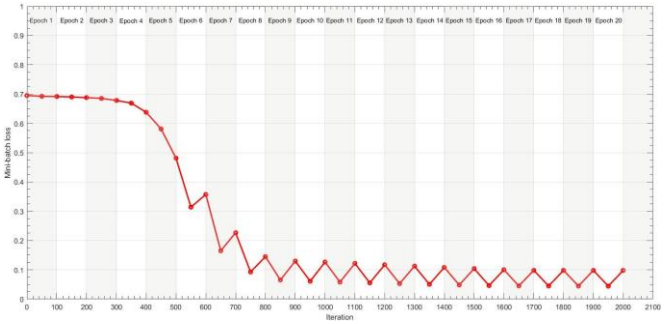


Figure 8 Mini-batch loss of the model training for 20 epochs.

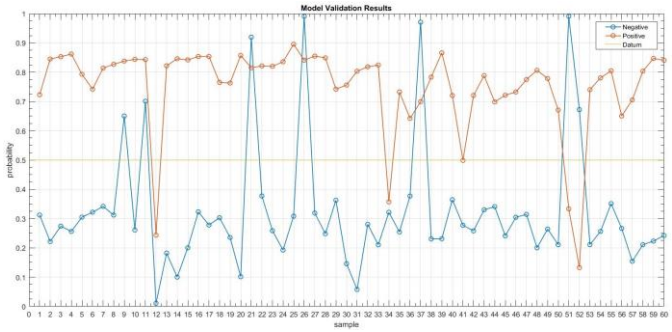


Figure 9 Model validation results of the COVID-19 positive and healthy test subjects.

A confusion matrix or also known as an error matrix is a representation of the performance of an algorithm. The confusion matrix is commonly used in the area of machine learning, typically supervised learning. The entries in the confusion metrics were calculated from the coincidence matrix by using the following hypothesis,

True Negative (TN) is the number of correct predictions that an instance is negative.

True Positive (TP) is the number of correct predictions that an instance is positive.

False Positive (FP) is the number of incorrect predictions that an instance is positive.

False Negative (FN) is the number of incorrect predictions that an instance is negative.

Figure 10 illustrates the confusion matrix of the results. For the development of the confusion matrix, the following parameters were identified by considering actual and observed values.

True Positives (TP) : 56

True Negatives (TN) : 53

False Positives (FP) : 04

False Negatives (FN) : 07

Equations 1 and 2 are the accuracy and precision of the test results based on the data extracted from the confusion matrix.

$$Accuracy = \frac{\sum(TP + TN)}{\sum(TP + FP + FN + TN)} \quad (1)$$

$$Precision = \frac{\sum(TP)}{\sum(TP + FP)} \quad (2)$$

N=120	**Actual: Yes**	**Actual: No**	
Predicted: Yes	TP = 56	FP = 04	60
Predicted: No	FN = 07	TN = 53	60
	63	57	

Figure 10 Confusion matrix of the test results.

Table 3 Accuracy, precision and recall values of the results

Class	n (truth)	n (classified)	Accuracy	Precision	Recall	F1 Score
1	63	60	90.83 %	0.93	0.89	0.91
2	57	60	90.83 %	0.88	0.93	0.91

According to the Table 3, the accuracy of the proposed design was 90.83% with 0.91 of average precision. Precision for both classes (class 1 and class 2) were 0.93 and 0.88 observed.

$$MAE = \frac{\sum_{t=1}^{N} |Y_t - F_t|}{N} \quad (3)$$

Mean Absolute Error (MAE) is a calculation of errors between paired measurements that express the same phenomena. Equation 3 represents the relationship between real data and the prediction data. The best Mean Absolute Error of a system is considered to be less than 0.200 and the MAE of the proposed system was 0.095. Therefore, the Lower MAE validates the accuracy of the proposed model for the identification of COVID-19 using chest CT scan images. Root Mean Square Error (RMSE) of the system was 0.149 calculated. Lower RMSE suggested the higher accuracy of the proposed algorithm.

V. Conclusion

Coronavirus disease outbreak 2019 (COVID-19) is a worldwide epidemic that has a significant effect not only on the health of peoples but also on the global economy. Pneumonia caused by coronavirus reveals a common hazy spot on the outer edges of the lungs, which indicates a trend such that machine learning methods can be used for early coronavirus identification.

In this paper, we addressed the role of artificial intelligence (AI) techniques in identifying the novel COVID-19 using CT chest scan images of corona patients. Training and testing were carried out using the dataset published by Ning and his research team at the Huazhong University of Science and Technology in China. As the model based on the CT scan showed good results in terms of accuracy and as it took less time to identify the COVID-19. The results show that the detection of COVID-19 is possible using the CT images with deep neural network methodology. The proposed algorithm was 90.83% accurate with the 0.095 Mean Absolute Error (MAE). Moreover, patients with lung disorders are identified as positive cases of COVID-19. As a result, an error may be developed in the other lung condition that has hazy regions of the lungs. The probability of the CT image result indicates COVID-19 detection at what level. Higher probability means larger hazy spots detect in the lungs which means the subject has a higher level of COVID-19 virus.

This study found some drawbacks, such as validation data of the CT dataset collected from one geographical region, which may not be representative of all COVID-19 patients in other geographic areas. In our future work, we will extend the algorithm to quantify the severity of other pneumonia using transfer learning and to validate the results using data obtained from many geographical regions of the world.

References

[1] Cheng, V.C.C., Lau, S.K.P., Woo, P.C.Y., Yuen, K.Y.: Severe Acute Respiratory Syndrome Coronavirus as an Agent of Emerging and Reemerging Infection. Clinical Microbiology Reviews. 20, 660-694 (2007)

[2] Ramadan, N., Shaib, H.: Middle East Respiratory Syndrome Coronavirus (MERS-CoV): A Review. Germs. 9, 35-42 (2019)

[3] Kamradt-Scott, A.: A Public Health Emergency of International Concern? Response to a Proposal to Apply the International Health Regulations to Antimicrobial Resistance. PLoS Medicine. 8. 1-5 (2011)

[4] Punn, N.S., Sonbhadra, S.K., Agarwal, S.: COVID-19 Epidemic Analysis using Machine Learning and Deep Learning Algorithms. MedR. XIV. 1-10 (2020)

[5] Jiang, F., Deng, L., Zhang, L., Cai, Y., Cheung, C.W., Xia, Z.: Review of the Clinical Characteristics of Coronavirus Disease 2019 (COVID-19). Journal of General Internal Medicine. 35. 1545-1549 (2020)

[6] Jin, Y.H., Cai, L., Cheng, Z.S., Cheng, H., Deng, T., Fan, Y.P.: A rapid advice guideline for the diagnosis and

treatment of 2019 novel coronavirus (2019-nCoV) infected pneumonia (standard version). Military Medical Research. 7, 1-23 (2020)

[7] Farias, L., Fonseca, E., Strabelli, D., Loureiro, B., Neves, Y., Rodrigues, T., Chate, R., Nomura, C., Sawamura, M.V.Y., Cerri, G.: Imaging findings in COVID-19 pneumonia. Clinics (Sao Paulo, Brazil). 75, 1-9 (2020)

[8] Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., Yu, L., Chen, Y., Su, J., Lang, G., Li, Y., Zhoa, H., Xu, K., Ruan, L., Wu, W.: Deep learning system to screen coronavirus disease 2019 pneumonia. arXiv. 2020, 1-29 (2020)

[9] Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z. and Shi, Y.: Lung infection quantification of COVID-19 in CT images with deep learning. arXiv. 2020, 1-19 (2020)

[10] Sachin, S.: Drawing insights from COVID-19-infected patients using CT scan images and machine learning techniques: a study on 200 patients. Environmental science and pollution research international. 27, 1-29 (2020)

[11] Harmon, S.A., Sanford, T.H., Xu, S.: Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. Nat Commun. 11, 1-7 (2020)

[12] Xavier, B.: Computer-aided covid-19 patient screening using chest images (X-Ray and CT scans). medRxiv. 2020, 1-19 (2020)

[13] Schmitt, W., Marchiori, E.: Covid-19: Round and oval areas of ground-glass opacity. Pulmonology. 26, 1-5 (2020)

[14] Ning, W., Lei, S., Yang, J., Cao, Y., Jiang, P., Yang, Q., Zhang, J., Wang, X., Chen, F., Geng, Z., Xiong, L., Zhou, H., Guo, Y., Zeng, Y., Shi, H., Wang, L., Xue, Y., Wang, Z.. iCTCF: an integrative resource of chest computed tomography images and clinical features of patients with COVID-19 pneumonia. Europe PMC. 2020, 1-37 (2020)

On Computing Memory as a Result of Processing

Chinthanie Weerakoon[1], Asoka Karunananda[2], Naomal Dias[3]
[1]Department of Statistics & Computer Science, University of Kelaniya, Dalugama, Sri Lanka.
[2]Department of Computational Mathematics, University of Moratuwa, Katubedda, Sri Lanka.
[3]Department of Computer Systems Engineering, University of Kelaniya, Dalugama, Sri Lanka.
email : [1]chinthanie@kln.ac.lk, [2]asokakaru@uom.lk, [3]ngjdias@kln.ac.lk

***Abstract*—It was difficult to find a computing model that has been constructed by imitating an Eastern-Philosophical-Approach-based human mind model to improve the computational efficiency. In this context, introducing a computing model that displays the features of the human mind with its evolving memory and has the ability to improve the processing power in subsequent program execution cycles, was a great research challenge. The Six-state Continuous Processing Model was proposed to fill this gap. This paper presents an extended research work on this model. Further, this new model has been compared with different computing models. As the basis for these comparisons, it takes the implementation and the execution of Quicksort algorithm. In this regard, each of these existing computing models have used different empirical settings. Therefore, the proposed model was compared with these computing models by using different experimental setups and conducting the experiments accordingly and separately. When conducting the experiments, it could identify different ranges of inputs, and experimental setups that enable this model to show better performance than the currently existing models. In some experimental scenarios, the performance improvement of the proposed model was more than 80% with compared to the other computing models.**

Keywords—Continuous Processing; Evolving Memory; Smaller Tactics Memory; Computational Efficiency; Human Memory; Conditional Phenomena; Memory as a Result of Processing.

I. INTRODUCTION

Von-Neumann introduced the computer architecture where memory is separated from the processor [1]. This architecture has been practiced in developing computers with various memory power and processing power. However, the demand for computers with high memory and processing power remains a challenge. In this context, while many hardware solutions including high speed memories (RAM, Cache [2], Registers) and processors (multi cores, GPU) [3] have been introduced, the development of new software technologies to use power of such hardware has been rather insignificant, as non of these software level introductions are still efficient to the expectation[4]. Therefore, finding a new computing model to enhance the computational efficiency has been a continuous research challenge.

This research has identified that the human mind is still the best computer which can generate better solutions over subsequent execution cycles of same program in a shorter period [5]. This has not been the case in computation on Von-Neumann machine. The study in [5] also revealed that mind is quite different from Von-Neumann computer, because the mind has no separate units as memory and processor, but memory is a result of continuous processing [6] in the mind. Mind operates as a continuous flow of thoughts [7] pertaining to an input or conditions. Thought flows can occur with respect to inputs coming from five sensors as well as from the mind itself [7]. The way mind react on an input is dependent on the past state of the mind. For instance, reactions to similar type of inputs inertia will be developed. This memory evolves over the cycles and aids the processing efficiency and accuracy. Therefore, as inspired from the Eastern Philosophical Approach (EPA) [38], we postulate that a computer with evolving memory[8] can be modelled by developing a processing model leading to software solution for improving computational efficiency and accuracy[5].

Knowing the fact that processing speed is dependent on the states and actions to manage the processors, we have critically studied various memory and processing models in both the hardware and softtware level. Most of these introductions were based on the memory and the processor sepretation. When it came to hardware level, there were different processor and memory models and connective mechanisms that were presented to improve computing power. In fact, different processors with different speedups have been invented with the time. For example, so far the world's fastest processor [9] is Intel's core i9 and also it is the best gaming processor [10]. Intel could do this, because of the new chipset. i.e. X299 [9]. Further, it has been expanded its processing power from eight core i9-9900 to eighteen core i9-9980XE [9]. These were achieved through different hardware formations such as increasing the dencity and reducing the size of chips [11]. Further, to support processing, various memory models were introduced, such as RAM, SRAM, DRAM, SDRAM. However, together with the bandwidth-wall, the disparity of speed between the CPU and the memory that is called as the memory-wall [12] degrades the performance of computing. Therefore, the separation between the memory and the processing is evident and this separation is one of the major obstacle in performance gaining in the computer at hardware level. Then, the other obstacles in gaining the performance of the computer is insufficient software level improvements to cope with underline architecture, and their own separated memory and processing models.

Further, this study revealed that, the most of the existing software solutions were mainly focused on providing solutions for the real world problems. These problems are arising from the natural systems with large number of entities that are connected each other and operated in distributed or parallel manner in the environments, which is changing dynamically. However, providing quality solutions more efficiently over subsequent generations of system executions were considered minimally. Those would be rather the

modelling of real world systems and finding best solutions, where the focus was sometimes bit deviated from enhancing the computational efficiency. This research has narrowed down its literature review to analyze the memory and processing in computing models such as Incremental Computing, Genetic Programming, and Multi-Agent systems. Multi Agent Systems (MAS) has been involved in problem solving by sending messages among a group of agents [13] having inspirations from the behavior of natural complex organizations [14] such as ant colonies, fish schools, and bee colonies. MAS offered a novel model for distributed and parallel computing on VNA and can yield emergent solutions [15]. However, when the certain applications of MAS are dealing with large groups of agents working together in the same stage, the efficiency improvement in such system cannot be expected [14]. In fact, the concepts such as logical agents[16], long short term memories and reinforcement learning[17], were quite impressive. Many of those had insights from the human mind according to the theories introduced in Bartlett's Remembering [18], constructive memory [19], and Atkinson-Shiffrin [20] and the Baddeley [21] models with a western philosophical view. Meanwhile, the foundation of the evolutionary computing was laid by the Genetic Algorithms (GA), having inspirations from the Darwinian theory of evolution [22]. There are many aspects in the field of computing that are benefited from GA. For example, for CPU scheduling GA has been applied [23] in order to maximize CPU throughput or utilization [24] or optimize the waiting time [25]. Further, over generations of executions, the GA can produce better quality solutions, although the GA consumed memory and CPU in a considerable level. The Incremental computing was an approach in modeling systems with the incremental and dynamic slight changes in input data [26]. There, the memory management [27] was done through the graphs and memorization [28]. Self-adjusting computing [29] was one of the branches in incremental computing. Further, there were different adaptive algorithms that have been applied in order to speed up the computing [30], even using GA [31]. In some cases, it has also been used different program transformation [32] techniques to enable adaptability and achieve speeding up. Further, it has been discussed memory and processing in parallel computing and neural computing also.

Finally, it has been come up with six-state processing model (SSPM) to develop the said evolvable memory. This processing model involves a set of special actions together with an extended Ready state [33] than traditional ready state, and a new Sleep state and a Terminate state that are deviated from traditional Exit state. Identification of new states and actions are based on the EPA. The SSPM has been formally validated for Turing Machine compatibility [37] and tested for some real-world problem solving. The results show that new processor model has been able to aid evolution of the memory and improved efficiency and accuracy in processing, with continuous processing. This model has been used to customize several existing programs such as Fraction Calculator (FC), QuickSort, and Quadretic Equation Solver (QES) introducing SSPM-FC, SSPM Sorting (SSPM Insertion, and SSPM-S-Equal), and SSPM-QES, in which the new model is incorporated, has been executed for many rounds with sets of inputs. Meanwhile, the time taken for the computation of each input has been recorded in nanoseconds. All the cases were tested to check whether an improvement has gained by creating modules for frequent operations over program execution cycles. The time values were collected for the execution of each equation in the same set of equation before and after do the modification. Then, the paired samples of time values were statistically analyzed with the paired-t-test after checking the samples for the applicability of the test in the samples. Finally, with the SSPM-FC, SSPM-QES, and SSPM-S-Equal, it could prove that the system gain improvement over generations of program executions by generating modules for frequent operations. This paper focuses on the work related to SSPM-Sorting.

The rest of the article is organized as below. The Section 2 explains th proposed model, whereas the Section 3 discusses the research methodology, while Section 4 reports the results and discusses the findings. Finally, the last section concludes the work.

II. PROPOSED MODEL

This research hypothesized that the processing power of the computer can be enhanced with the support of a smaller tactics memory, which improves as a result of continuous processing. This section proposed the six-state continuous processing model, and it is the core of this thesis. The model is abbreviated as SSPM. The SSPM system initially begins by an internal process. Then, the particular operation for the process has been arbitrarily picked out from the bunch of operations that are stored in the initial smaller tactics memory. Further, the instructions saved in the knowledgebase can be executed through this smaller memory. The system shifts to the internal mode once an internal input is entered. The system can receive an external input only when the present inner process sleeps. After moving to the external mode, if there is no external input, the system can move back to the inner process. However, if there is no external input, the inner process can be proceeded with the actions linked to the latest external input. The system conducts ongoing processing in such a manner.

The model accomplishes a series of tasks, during the ongoing processing over generations. Particularly, it identifies the inputs and operations, adds library files for new operations, classifies appropriate operations with respective information and directives, prioritizes the operations relevantly, creates recurrently arising operating modules and deletes needless or wasteful modules and directives as well as the useless information. Such a way, the corresponding entries develop and organizes the smaller memory. In addition to that, these actions under the above mentioned two process categories can occur in a one stream. Moreover, the tasks, namely, deletion, classification, additions, and prioritization can be accumulated under the general term 'Organizing'. Consequently, the system is gaining improvements by iterating this organizing job. Depending on the process category, the results of each process can also be generated externally or internally.

The newly presented computing model comprises of six states, specifically "New", "Ready", "Running", "Blocked", "Sleep", and "Terminate" as shown in figure 1.

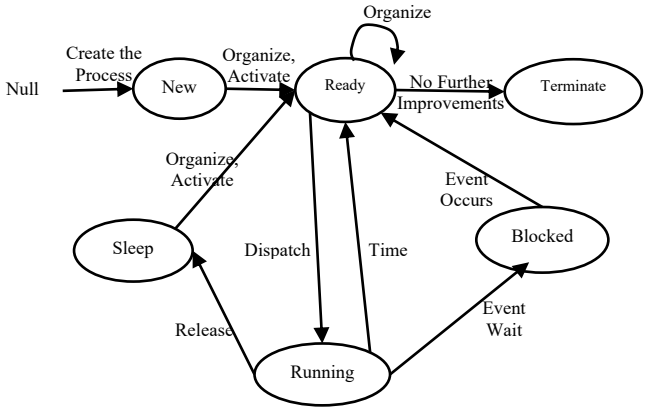


Figure 11 Six-state Continuous Processing Model (SSPM)

At first, neither the recently generated processes were organized nor activated those processes were in the 'New' state. Once the processes were organized and activated, those were moved to the 'Ready' state. Then, a process, which was running on the processor was in the 'Running' state. A process was switched to the 'Sleep' state after finishing the execution enabling some other process to be initiated, executed or continued. Furthermore, if a process had to wait until a specific task to be completed, then the process was in 'Blocked' state. Finally, if a process was neither necessary to be modified nor requires any execution can be ended, and the state can be updated as 'Terminate'. The states and movements between the states in the novel computing model are perceptibly illustrated in the figure 1. Introducing this model, it was expected to enhance the processing in the system and the memory process using a set of tactics maintaining the continuity. The coming section describes the exhibited features of the suggested model.

III. Research Method

This research was conducted to discover a new computing model to enhance the processing power of the computer. There, the characteristics of the human mind was incarnated in to a new processing model exploiting the EPA. There the mind a continuous flow of thoughts [34]. The continuity of this processing is maintained by several factors such as the inputs receive through physical five sense doors (external inputs), the inputs internally generate in the mind door (internal inputs), and a set of causal relations [5]. All the time, the internal inputs are generated in relation to and are affected by the prior external or internal inputs. In addition to that, the repeated processing on the same set of inputs, improve the speed, quality and the accuracy of processing [7]. There, the processing is not separated from the memory. The memory is a result of continuous processing that arise as per the conditions. Further, starting from an initial setup, the smaller tactics memory has gradually improved and organized through this continuous processing or practice. The knowledge and instruction entities entered in as any sort of inputs or instructions, are labeled, in the way, which one can identify, describe, relate or retrieve back the knowledge entities and the results of relevant computations. Moreover, a set of tactics such as pattern identification, classification, and prioritization has been used.

An inspirational example that has displayed the nature of the human mind is discussed next. Let's think about the two cases, where a student, and a senior professor who are preparing for and do their presentations. When, the student does the presentation, in most of the cases, he needs external aids such as power point slides to drive through his own knowledgebase. Through series of refinements, he can well organize his slides using set of tactics and improve his own ability to do the presentation accessing his knowledgebase. In the case of a senior professor, he has such a well-organized smaller tactics memory, which enables him to clearly conduct his presentation accessing his larger knowledgebase. This ability and the smaller tactics memory has been improved throughout the years. All such skilled workers do in the same way. Therefore, one can believe the existence of a smaller tactics memory in the human mind. This smaller tactics memory gradually updates through continuous processing, and allows access to the large knowledgebase, is a part of processing, and improves the processing back. Again, it is obvious that this smaller tactics memory has been different from the smaller memories of the current computer such as caches or registers [19]. Through this continuous processing, human can improve the processing power, accuracy and the quality of the work they do. Then, this would be a new approach for computing to improve computational efficiency. This processing model can improve the processing power, quality, and accuracy of the computation done by the computer with the support of an evolving smaller memory, which is a result of continuous processing.

In software level, it is an evident fact that the efficiency of processing hugely affected by the actions and the corresponding states in the process flow. After critically studying memory and processing models as mentioned earlier, this research has introduced SSPM [5] to produce the conditionally evolving smaller memory. The actions, the constituent of the transitions of the new processing model were formed by utilizing the set of tactics have been derived from a set of fifteen causal relations, namely, Object, Root, Co-Nascence, Association, Mutuality, Pre-Dominance, Presence, Support, Pre-Nascence, Proximity, Karma, Repetition, Disappearance, Post-Nascence, Karma-Result from twenty-four causal relations [5].

This model with the above characteristics and the actions has been incorporated in to a Fraction Calculator (SSPM-FC) [5], quicksort algorithm (SSPM Sorting – particularly SSPM Insertion, and SSPM-S-Equal), quadratic equation solver (SSPM-QES), and in a simulated process scheduling program (SSPM-PS). However, it has been done a great work on FC as it has matched better with these conditions and circumstances of the proposed model than the other systems. SSPM-Sorting allowed to compare the model with existing computing models. The implementation of this SSPM embedded sorting program was based on Quicksort. The Quicksort is highly efficient algorithm for sorting and is based on partitioning the data set into two subsets. There, the original list of elements is divided into two sub lists, one of which holds values lower than a selected particular value, the pivot, depending on which the division is made and the other list holds values higher than the pivot value.

Further, it was concentrated on comparing this system with the quicksort programs developed in some other computing models such as parallel computing, ADAPTON, Self-Adjusting computing, Dynamically Tuned Library and Evolutionary computing. This development process was also similar to that of the SSPM-FC [5], except its Input-Content Analyzer, internal input creation and the calculation. However, it had to do comparisons with the lists of larger number of inputs, it has been used internal process case more. Specially, in the scenario, which was tried to compare the model with the Dynamically Tuned library and the Evolutionary computing, the standard deviation and the distribution of data were also mattered. Therefore, when creating the internal scenario rather than creating lists with random numbers it was required to write the code so to create normally distributed data with fixed standard deviation. In contrast to the FC and QES, The SSPM-Sorting program has been implemented so to support all the input patterns mentioned below. Further, this had the techniques such as Insert for IP1, Equal for IP4, delete for IP3, and sort for IP2.

$$IP1:\ X \subseteq Y = X$$

$$IP2:\ X \not\subseteq Y,\ Y \not\subseteq X \Rightarrow X \cap Y \neq \phi \ or\ X \cap Y = \phi$$

$$IP3:\ Y \subseteq X = Y$$

$$IP4:\ X = Y$$

The overall hypothesis of this research has reduced to the following words.

The hypothesis, which was tested in this scenario is;

H_0: There is no difference between the means of time values before and after organizing the tactics memory by applying modifications through continuous processing.

(No Performance Improvement over program

Table 1 Comparison Tables (a) Average run times for different thresholds and number of elements for parallel QS (Source: [35]), (b) Relevantly tested SSPM, sorting list results with original QS, when there are 1, half and all new, all equal elements than/to previous list in SSPM

(a) No of elements	**T=1000 (ms)**	**T=5000 (ms)**	**T=50000 (ms)**
10	0.01	0 0.01	0001
100	0.01	0.020001	0.050004
1000	0.250016	0.270011	0.260018
10000	2.010118	2.880166	3.060169
25000	5.380318	6.120344	9.15052
50000	11.36065	11.320644	19.61112
75000	18.14103	18.251045	28.60164
100000	24.91142	22.591294	34.19196
150000	36.41208	34.551976	46.99269

(b) No of Elements	**QS (ms) (Original)**	**SSPM (Ins) Sorting (ms) (New-1)**	**SSPM (Ins) Sorting (ms) (New-All)**	**SSPM (Ins) Sorting (ms) (New-Half)**	**SSPM (Equ) Sorting (ms)**
10	0.066632	0.018038	0.038061	0.039192	0.003218
100	0.539753	0.07334	0.348982	0.243148	0.002344
1000	3.739901	1.930715	3.801648	2.393156	0.013139
10000	55.54588	170.6803	30.75075	58.83381	0.0598
25000	110.9843	235.1399	105.149	467.4121	0.336187
50000	2.85E+08	9.76E+08	260.1235	1514.95	0.542807
75000	467.8586	2011.655	449.4495	3625.402	0.597209
100000	678.7099	4018.923	719.3369	7379.43	1.139867
150000	1534.874	8671.315	1338.664	16466.74	1.436503

execution cycles) (H_0: $\mu_D = d_0$, d_0=0)

H_1: The mean value of the time values collected before organizing memory is greater than the mean value of the time values collected after organizing memory.

(Performance has improved over program execution cycles) (H_1: $\mu_D > d_0$, d_0=0)

For each of the sub scenarios in testing of SSPM-Sorting, 75 lists of 100 integers in-between 0 and 1000 have been randomly generated. For examples, to check InsCalcModule, a set of 75 lists of 100 integers have been used before and after create the module. Then, those list have been sorted before the modification and collected the time taken by each expression for the execution in Nano seconds. In the same way, after the modification, the time values have been collected for the same set of expressions in Nano seconds. This collection process has been conducted in subsequent sorting cycles. It is an important fact to remind that the entire processing model in the above mentioned programs are managed through a smaller memory with the set of tactics. Finally, all these were tested with many examples and evaluated. The next section has briefly mentioned the respective testing, results and comparisons that have been taken place.

IV. Results and Discussion

By applying paired-t test with 99% confidence level under the Testing Scenario 1 and 95% confidence interval under Testing Scenario 2, it has been able to prove that by customizing the FC with the proposed continuous processing model, helps to improve the performance of an FC, when executes over the generations. Further, it could prove that the SSPM-QES can gain improvement over subsequent execution cycles, similarly applying paired-t test with 95% confidence interval.

The program SSPM-Sorting was tested under five testing scenarios. Under the first (SSPM-S-Insertion) and the second (SSPM-S-Equal), it could prove that the system can gain improvement over consecutive program executions, by applying paired-t test with 95% confidence interval as similar to the above. Even though, the SSPM-S-Equal shows improvement for any total number of elements in the list, the performance of the SSPM-S-Insertion depends on the number of new elements and the total number of elements in the list with compared to the previous list. Next three testing scenarios were allocated to compare the model with the parallel computing [35] (Table 1), incremental computing [36] (Table 3), Self-adjusting computing [29] (Table 4), Dynamically Tuned Library (DTL) [30] for Sorting and Sorting with Genetic Algorithmic Approach (GAA) [31] (Table 4). The respective results are summarized in each section.

A. Compare with Parallel Computing

This section compares the speedup of the SSPM-Sorting with the speedup of the parallel Quicksort [35]. (Here the speedups are calculated with respect to the original quicksort algorithm). Here, it has randomly generated nine lists with the number of elements: 10, 100, 1000, 10000, 25000, 50000, 75000, 100000, and 150000. Then, the time taken for each sorting has recorded in milliseconds, before and after the modifications. The obtained values are recorded as in Table 1.

B. Compare with Self-Adjusting Computing and Incremental Computing

This This section compares the SSPM sorting with the Quicksort with self-adjusting computing [29] and the incremental computing [36]. The testing results obtained by the respective researchers have been compared here with the results obtained through executing different SSPM-Sorting techniques as seen in the tables table 2 and 3.

Case 1: Comparison with Self adjusting sort.

In this scenario, all the tests had used lists with total number of elements 100,000 as the input and compared the speedup gained by those compared to the original quicksort as seen in the table 1.

Table 2 Quicksort with Self-adjusting computing [29] Vs SSPM sorting

Sorting Technique	**Size of the list**	**Speedup**
Quicksort with Self-Adjusting Computing	$1*10^5$	654.06
SSPM-S-Insertion (90% new)	$1*10^5$	1.218861
SSPM-S-Insertion (95% new)	$1*10^5$	2.288098
SSPM-S-Insertion (100% new)	$1*10^5$	2.409908
SSPM-S-Equal	$1*10^5$	595.45

Case 2: Comparison with Incremental computing sort.

Under this, it has been considered four approaches, which have been upgraded with the incremental sorting. Here also, the size of the lists used consist of 100,000 elements. In addition to the speedup gained, the utilized maximum heap size used for the comparison is shown in the table 3 below.

Table 3 Quicksort with ADAPTON [30] with incremental computing Vs SSPM sorting

Sorting technique	**Size of the list**	**Speedup**	**Maximum utilized heap size (MB)**
Quicksort – LazyBidirectional-Eager	$1*10^5$	21600	162
Quicksort – LazyBidirectional-Lazy	$1*10^5$	2020	162
Quicksort – EagerTotalOrder - Eager	$1*10^5$	245	2680
Quicksort – EagerTotalOrder - Lazy	$1*10^5$	22.9	2680

SSPM-S-Insertion (90% new)	$1*10^5$	1.218861	2626
SSPM-S-Insertion (95% new)	$1*10^5$	2.288098	3127
SSPM-S-Insertion (100% new)	$1*10^5$	2.409908	1347
SSPM-S-Equal	$1*10^5$	595.45	2144

C. Compare with DTL and GAA Sorting

This testing scenario compares the SSPM-Sorting with the Dynamically Tuned Library (DTL) [30] for Sorting and Sorting with Genetic Algorithmic Approach (GAA) [31] was complicated than all the tests conducted so far. The DTL research suggested that the characteristics of input data and some architectural features affect the sorting. Particularly, the distribution of data, standard deviation, number of elements in the list, size of the cache, size of the cache line, and number of registers are among the factors. First, six lists of Normally distributed 2M (M=2^20) elements have been created. Each list was created so as to have a single standard deviation (stdv) for all the elements in each list, where those nine stdvs were {100, 1000, 10000,100000, 1000000, and 10000000}. Same testing scenario has been conducted in two different computers: Intel(R) Xeon(R) CPU ES-2623 V3 @ 3.00 GHz with Turbo Boost up to 2.0GHz with 16GB cache size, 64B cache line size and 4 registers in SUSE Linux (Server), and Intel(R) Core i7-8550U 1.8GHz with Turbo Boost up to 4.0GHz with 4608MB cache size, 64B cache line size and 8 registers in Windows 10 operating system (Laptop). There is an apparent speedup gain in the SSPM-Sorting for the lists with a standard deviation approximately less than 1000.

Then, again the SSPM-Sorting has been compared with the different improvements gained by the Quicksort after applying different adapting techniques through Dynamically Tuned Library (DTL) and a Genetic Algorithmic Approach (GAA) for sorting (Gene-Sort) [31] in an Intel PIII Xeon computer with 512KB cache size in RedHat 7.3 Operating System as seen in Table 4.

Table 4 Comparisons of SSPM sorting with DTL[30] and GAA [31] sorting

Sorting Technique	**Speedup**
DTL – Insert Sort at the end	1.1173
DTL – Insert Sort at each partition	1.0465
DTL – Sorting Networks	1.1672
GAA – Gene Sorting	2.5714
SSPM-S-Insertion (90% new) (Server)	3.25556
SSPM-S-Insertion (95% new) (Server)	6.30319
SSPM-S-Insertion (100% new) (Server)	8.63834
SSPM-S-Insertion (90% new) (Laptop)	3.898003
SSPM-S-Insertion (95% new) (Laptop)	7.984114
SSPM-S-Insertion (100% new) (Laptop)	8.559002
SSPM-S-Equal (0% new) (Laptop)	653.64

V. CONCLUSION

The final target of this research was to develop a continuous processing model to improve the computing efficiency of the computer that leads to a new theory of computing. There, the computer memory was modeled as conditional phenomena, which enhance the efficiency of continuous processing over program execution cycles. Further, several real-world processes, which have exhibited the continuous processing and evolving nature of the human mind, have rooted the research idea for improving the computing power. And it was a fantastic idea. SSPM, the model introduced, consists of three features, such as two processes (internal and external), continuous processing, and conditionally evolving memory. Further, the processing states of the proposed processing model were new, ready, running, blocked, sleep and terminate. With these states, set of actions forms the continuation of processing. Furthermore, the new model is advanced than the incremental computing, since the new model refine the entire system through a continuous process, not only the parts related to the modified input. On the other hand, 'Repetition' and 'Classification' can be shown as the major concepts.

REFERENCES

[1] M. Abd-El-Barr and H. El-Rewini, *Fundamentals of computer organization and architecture*. Hoboken, N.J: Wiley, 2005.

[2] W. J. Starke *et al.*, "The cache and memory subsystems of the IBM POWER8 processor," *IBM J. Res. Dev.*, vol. 59, no. 1, pp. 3:1-3:13, Jan. 2015.

[3] D. A. Patterson and J. L. Hennessy, *Computer organization and design*, 5th ed. oxford: Morgan Kaufmann Publishers, 2014.

[4] G. E. Moore, "Moore's Law at 40. Chapter 7," in *Understanding Moore's Law: Four decades of innovation edited by D. C. Brock*, Philadelphia, PA.: Chemical Heritage Foundation, 2006, pp. 67–84.

[5] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "Six-state Continuous Processing Model for a New Theory of Computing," presented at the SLAAI International Conference on Artificial Intelligence, University of Moratuwa, 2019, vol. 890, pp. 32–48.

[6] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "A tactics memory for a new theory of computing," 2013, pp. 153–158.

[7] B. Bodhi, *Comprehensive Manual of Abhidhamma: The Psychology of Buddhism (Abhidhammattha Sangaha)*. Buddhist Publication Society, 2006.

[8] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "Conditionally evolving memory for computers," 2015, pp. 271–271.

[9] Mihir Patkar, "Intel Core i9 vs. i7 vs. i5: Which CPU Should You Buy?," *Technology Explained*, 06-Dec-2018. [Online]. Available: https://www.makeuseof.com/tag/intel-core-i9-vs-i7-vs-i5-cpu/.

[10] Chaim Gartenberg, "Intel announces its latest 9th Gen chips, including its 'best gaming processor' Core i9," *Circuit Breaker*, 08-Oct-2018. [Online]. Available: https://www.theverge.com/2018/10/8/17950968/intel-9th-gen-core-chips-2018-desktop-processors-8-core-i9-9900k.

[11] W. Stallings, *Computer organization and architecture: designing for performance*. Upper Saddle Rive, N.J.: Pearson Prentice Hall, 2006.

[12] [12] S. A. McKee and others, "Reflections on the memory wall.," in *Conf. Computing Frontiers*, 2004, p. 162.

[13] [13] M. J. Wooldridge, *An introduction to multiagent systems*, 2nd ed. Chichester, U.K: John Wiley & Sons, 2009.

[14] N. Siddique and H. Adeli, "Nature Inspired Computing: An Overview and Some Future Directions," *Cogn. Comput.*, vol. 7, no. 6, pp. 706–714, Dec. 2015.

[15] J. Liu and K. C. Tsui, "Toward nature-inspired computing," *Commun. ACM*, vol. 49, no. 10, pp. 59–64, 2006.

[16] S. Costantini, "Defining and Maintaining Agent's Experience in Logical Agents," in *Informal Proc. of the LPMAS (Logic Programming for Multi-Agent Systems) Workshop at ICLP 2011, and CORR Proceedings of LANMR 2011, Latin-American Conference on NonMonotonic Reasoning*, Mexico, 2011, pp. 151–165.

[17] B. Bakker, "Reinforcement learning with long short-term memory," in *Advances in neural information processing systems*, 2002, pp. 1475–1482.

[18] F. C. Bartlett, F. C. Bartlett, and W. Kintsch, *Remembering: A study in experimental and social psychology*, vol. 14. Cambridge University Press, 1995.

[19] D. L. Schacter, "Constructive memory: past and future," *Dialogues Clin. Neurosci.*, vol. 14, no. 1, p. 7, 2012.

[20] Kenneth J. Malmberg, Jeroen G. W. Raaijmakers, and Richard M. Shiffrin, "50 years of research sparked by Atkinson and Shiffrin (1968)," *Mem. Cognit.*, vol. 47, no. 4, pp. 561–574, 2019.

[21] A. Baddeley, "Working memory," *Curr. Biol.*, vol. 20, no. 4, pp. R136–R140, 2010.

[22] K. A. De Jong, *Evolutionary Computaion: A Unified Approach*. London, England: The MIT Press, Cambridge, Massachusetts, 2006.

[23] M. Sharma, P. Sindhwani, and V. Maheshwari, "Genetic Algorithm Optimal approach for Scheduling Processes in Operating System," *Int. J. Comput. Sci. Netw. Secur.*, vol. 14, no. 5, pp. 91–94, 2014.

[24] Sindhwani P. and Wadhwa V., "Genetic algorithm Approach for Optimal CPU Scheduling," *IJCST*, vol. 2, no. 2, pp. 92–95, Jun. 2011.

[25] M. U. Siregar, "A New Approach to CPU Scheduling: Genetic Round Robin," *Int. J. Comput. Appl.*, vol. 47, no. 19, pp. 18–25, Jun. 2012.

[26] M. Carlsson, "Monads for incremental computing," in *The seventh ACM SIGPLAN international conference on Functional programming*, 2002, pp. 26–35.

[27] Matthew A. Hammer and Umut A. Acar, "Memory Management for Self-Adjusting Computation," presented at the Proceedings of the the 2008 International Symposium on Memory Management, Tucson, Arizona, USA, 2008.

[28] U. A. Acar, G. E. Blelloch, and R. Harper, *Selective memoization*, vol. 38. ACM, 2003.

[29] U. A. Acar, G. E. Blelloch, M. Blume, R. Happer, and K. Tangwongsan, "An experimental Analysis of Self-Adjusting Computation," *ACM Trans. Program. Lang. Syst.*, vol. 32, no. 1, 2009.

[30] Xiaoming Li, Maria Jesus Grzaran, and David Padua, "A dynamically Tuned Sorting Lirary," Los Alamitos, Calif., 2004.

[31] Xiaoming Li, Maria Jesus Grzaran, and David Padua, "Optimizing sorting with genetic algorithms," New York, NY, USA., 2005.

[32] Y. A. Liu, "Efficient computation via incremental computation," in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 1998, vol. 1574, pp. 194–203.

[33] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "New Processing Model for Operating Systems," University of Kelaniya, 2016, p. 29.

[34] W. F. Jayasuriya, *The Psychology & Philosophy of Buddhism*. Onalaska, USA: pariyatti, 2016.

[35] A. H. Almutairi and A. H. Alruwaili, "Improving of Quicksort Algorithm Performance by Sequential Thread or Parallel Algorithms," *Glob. J. Comput.*

Sci. Technol. - Hardw. Comput., vol. 12, no. 10, 2012.

[36] M. A. Hammer, K. Y. Phang, M. Hicks, and J. S. Foster, "Adapton: Composable, demand-driven incremental computation," 2014.

[37] Weerakoon C, Karunananda A, Dias N (2019b) Formal verification of conditionally evolving memory. Int J Comput Eng Inf Technol 11(11):243–257

[38] Weerakoon, C., Karunananda, A. & Dias, N. Human-mind-inspired processing model for computing. Mind Soc 19, 237–256 (2020). https://doi.org/10.1007/s11299-020-00236-2

Taking Causal Modeling to a Next Level: Self-Modeling Networks Adding Adaptivity to Causality

Jan Treur
Social AI Group, Department of Computer Science, Vrije Universiteit Amsterdam
email: j.treur@vu.nl

***Abstract*— Causal modeling is an intuitive, declarative way of modeling. Due to the universal character of causality, in principle it applies to practically all disciplines. In spite of this seemingly very wide scope of applicability, there are also serious limitations and challenges that stand in the way of applicability. This concerns in particular cases where dynamics and adaptivity play a role. This paper addresses these challenges by exploiting the notion of self-modeling network that has been developed from a Network Science perspective. Adaptivity is obtained by adding to a given causal base network, a self-model which represents part of the base network's causal structure. Moreover, this construction can easily be iterated so that multiple orders of adaptation can be covered as well. This indeed takes causal modeling to a next level in more than one way. Therefore, in this way dynamics and adaptivity are also covered well, which substantially widens the scope of applicability of causal modeling.**

Keywords*— *causal modeling, self-modeling network, network reification, adaptive social network, controlled adaptation

I. INTRODUCTION

Causal modelling provides a declarative approach that has a long tradition in Artificial Intelligence; e.g., [1-4]. One of the challenges, however, is that causal modelling involving cyclic paths in causal graphs poses difficulties; therefore many approaches to causal modelling limit themselves to Directed Acyclic Graphs (DAG's). More in general, to avoid temporal complexity, dynamics is often not addressed in approaches based on causal networks, neither for the causal effects on nodes, nor for the network structure itself. The difficulty to allow cyclic paths in a causal network is one consequence of this form of abstraction from dynamics of the nodes in a causal network. Another consequence of abstracting from dynamics is that distinctions in timing and asynchrony of causal effects (i.e., how fast causal effects actually are effectuated) cannot be made, whereas often such differences in timing and asynchrony are crucial for real-world processes modelled by a causal network. Finally, within causal models, not only the nodes but also the causal relations are usually considered static, they cannot change over time. This excludes many adaptive real-world processes from the scope of applicability for causal modeling.

In the meantime, working from the perspective of Network Science, new approaches have been developed that can be used to overcome the above-mentioned limitations of causal modeling. In particular, in this paper it will be addressed how both within-network dynamics (dynamics of the node states) for causal network models and adaptivity of the causal relations can be addressed using the network-oriented modeling approach developed in [5-7].

Using this approach as introduced for within-network dynamics in [5], the dynamic perspective is based on a continuous time dimension, represented by real numbers, so that all nodes have state values (also represented by real numbers) that vary over time. The added temporal dimension enables modelling by cyclic causal networks as well, and also timing of causal effects can be modelled in detail and differently per node, so that also asynchronous processes are covered. Due to this, causal modeling can be used for causal networks that contain cycles, such as many networks modelling mental or brain processes, or networks describing social interaction processes (for example, in social media). Moreover, in [5, 7] it is shown how – supported by a dedicated software environment – networks with these within-network dynamics can be specified by declarative means, by mathematical relations and functions; the modeler does not need to address procedural descriptions nor program code.

In addition to these within-network dynamics, another useful element from the network-oriented modeling perspective is the notion of self-modeling network or reified network introduced in [6-8]. This is a network that includes a self-model for part of its own network structure in the form of nodes that represent certain network structure characteristics such as connection weights or excitability thresholds. Any (base) network can be extended by including such a self-model, which can be considered to be at a next level, compared to the base network; this step is also called network reification; e.g., [6-8]. This construction for networks in particular was inspired by another long-standing tradition in AI, namely that of meta-programming and metalevel architectures; e.g., [9-13]. Having such self-models within a network enables to model adaptation of the network structure by the within-network dynamics of the self-model representing this network structure. As the latter can be specified by declarative means in the form of mathematical relations and functions, also adaptivity of the network structure can be specified in a similar declarative manner. To support the modeler, a dedicated software environment (described in [7], Ch 9) is available that also applies to self-modeling networks.

In this paper, the perspective pointed out above will be illustrated in more detail. First in Section 2 the network-oriented modeling approach based on self-modeling networks will be briefly introduced. Next, in Section 3 it will be illustrated for an example of a multilevel second-order adaptive causal (social) network model for bonding

by (faked) homophily, while in Section 4 an example of a simulated scenario for this model is described. Finally, Section 5 is a discussion.

II. Modeling Adaptivity by Self-Modeling Networks

In this section, the network-oriented modeling approach by self-modeling networks used is briefly introduced in two steps.

A. Network-Oriented Modeling by Temporal-Causal Networks

As in this approach *nodes Y* in a network have activation values *Y(t)* that are dynamic over time *t*, they serve as state variables and will usually be simply called *states*. For these dynamics, the states are considered to affect each other by the connections within the network; therefore these connections are interpreted here as causal relations. This has been inspired partly by how in Philosophy of Mind networks of mental states and their causation relations are described; e.g., [14]. In line with this, following [5, 7], a basic *temporal-causal network structure* is characterised by:

- *Connectivity characteristics*
  Connections from a state *X* to a state *Y* and their weights $\boldsymbol{\omega}_{X,Y}$
- *aggregation characteristics*
  For any node *Y*, some combination function $\mathbf{c}_Y(..)$ defines aggregation that is applied to the impacts $\boldsymbol{\omega}_{X_i,Y}X_i(t)$ on *Y* from its incoming connections from states $X_1, \cdots, X_k$
- *timing characteristics*
  Each state *Y* has a speed factor $\boldsymbol{\eta}_Y$ defining how fast it changes for given causal impact

Here, the states X_i and *Y* have activation levels $X_i(t)$ and $Y(t)$ that vary (often within the [0, 1] interval) over time, described by real numbers *t*. These dynamics are described by the following difference (or differential) equations that incorporate in a canonical manner the network characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$:

$$Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \cdots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]\,\Delta t \quad (1)$$

for any state *Y* and where $X_1, \cdots, X_k$ are the states from which *Y* gets its incoming connections. The equations (1) are useful for simulation purposes and also for analysis of properties of the emerging behaviour of temporal-causal networks. The overall combination function $\mathbf{c}_Y(..)$ for state *Y* is taken as the weighted average of some of the available basic combination functions $\mathbf{c}_j(..)$ by specified weights $\boldsymbol{\gamma}_{j,Y}$, and parameters $\boldsymbol{\pi}_{1,j,Y}$, $\boldsymbol{\pi}_{2,j,Y}$ of $\mathbf{c}_j(..)$, for *Y*:

$$\mathbf{c}_Y(V_1, \ldots, V_k) = \frac{\boldsymbol{\gamma}_{1,Y}\,\mathbf{c}_1(V_1,\ldots,V_k) + \ldots + \boldsymbol{\gamma}_{m,Y}\,\mathbf{c}_m(V_1,\ldots,V_k)}{\boldsymbol{\gamma}_{1,Y} + \ldots + \boldsymbol{\gamma}_{m,Y}} \quad (2)$$

Such equations (1), (2) are hidden in the dedicated software environment that can be used for simulation and analysis; see [7], Ch 9. This software environment is freely downloadable from URL

https://www.researchgate.net/project/Network-Oriented-Modeling-Software.

Combination functions are similar to the functions used in a static manner in the deterministic Structural Causal Model perspective described, for example, in [3, 4, 15]. However, in the Network-Oriented Modelling approach described here they are used in a dynamic manner. For example, Pearl [3], p. 203, denotes nodes by V_i and combination functions by f_i (although he uses a different term for these functions). In the following quote he points at the issue of underspecification concerning aggregation of multiple connections, as in the often used graph representations the specification of combination functions f_i for nodes V_i, is lacking:

'Every causal model *M* can be associated with a directed graph (…) This graph merely identifies the endogeneous and background variables that have a direct influence on each V_i; it does not specify the functional form of f_i.' [3], p. 203

Therefore, in addition to graph representations for connectivity, at least aggregation in terms of combination functions has to be addressed, as indeed is done for temporal-causal networks, in order to avoid this problem of underspecification. That is the reason why aggregation in terms of combination functions is part of the definition of the network structure for temporal-causal networks, in addition to connectivity in terms of connections and their weights and timing in terms of speed factors.

As part of the software environment, a large number > 35 of useful basic combination functions are included in a Combination Function Library, and also a facility to easily indicate any function composition of any available basic combination functions in the library. One of the combination functions from this library used for states *Y* in the example network model described in Section 3 is:

the *Euclidean combination function* $\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k)$ defined by

$$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \sqrt[n]{\frac{V_1^{\,n} + \cdots + V_k^{\,n}}{\lambda}} \quad (3)$$

where n is the order and λ a scaling factor and $V_1, \ldots, V_k$ are the impacts from the states from which the considered state *Y* gets incoming connections.

In Section 3, it will be explained in more detail how the combination function $\mathbf{eucl}_{n,\lambda}(\ldots)$ is used to model social contagion. Social contagion makes that states of connected persons such as emotions or opinions, causally affect each other; e.g., (Levy and Nail, 1993).

The above concepts (the characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$) enable to design network models and their dynamics in a declarative manner, based on mathematically defined functions and relations for them. Note that for each state *Y*, all characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$ mentioned above causally affect the activation level of *Y*, as also can be seen from equations (1) and (2). Each of these characteristics do that causing in their own way from a specific role, either for connectivity, for aggregation or for timing. Below, this observation will also turn out useful in the context of self-models to address adaptivity.

B. Using Self-Modeling Networks to Model Adaptive Networks

Realistic network models are usually adaptive: often some of their network characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$ change over time. For example, for mental networks often the connections are assumed to change by hebbian learning [16] and for social networks, it is often assumed that connections between persons change through a bonding by homophily principle [17-19].

Adaptive networks are often modeled in a hybrid manner by considering two different types of separate models that interact with each other: a network model for the base network and its within-network dynamics, and a numerical model for the adaptivity of the network structure characteristics of the base network. The latter dynamic model is usually specified in a format outside the context of network modeling: in the form of some adaptation-specific procedural or algorithmic programming specification used to run the difference or differential equations underlying the network adaptation process.

In contrast, by including *self-models*, a network-oriented conceptualisation similar to what was described above, can also be applied to adaptive networks to obtain a declarative description using mathematically defined functions and relations for them as well; see [6, 7]. This works through the addition of new states to the network (called *self-model states*) which represent network characteristics by network states. Then the causal impacts of these characteristics on a state Y as mentioned above can be modelled as causal impacts from such self-model states. This brings the causal impacts from these characteristics on a state Y in the standard form of a causal model where via causal connections nodes affect other nodes.

More specifically, adding a self-model for a temporal-causal base network is done in the way that for some of the states Y of the base network and some of the network structure characteristics for connectivity, aggregation and timing (i.e., some from $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$), additional network states $\mathbf{W}_{X,Y}$, $\mathbf{C}_{j,Y}$, $\mathbf{P}_{i,j,Y}$, $\mathbf{H}_Y$ (*self-model states* or *reification states*) are introduced and connected to other states:

a) Connectivity self-model

- Self-model states $\mathbf{W}_{X,Y}$ are added representing connectivity characteristics, in particular connection weights $\boldsymbol{\omega}_{X,Y}$

b) Aggregation self-model

- Self-model states $\mathbf{C}_{j,Y}$ are added representing aggregation characteristics, in particular combination function weights $\boldsymbol{\gamma}_{j,Y}$
- Self-model states $\mathbf{P}_{i,j,Y}$ are added representing aggregation characteristics, in particular combination function parameters $\boldsymbol{\pi}_{i,j,Y}$

c) Timing self-model

- Self-model states $\mathbf{H}_Y$ are added representing timing characteristics, in particular speed factors $\boldsymbol{\eta}_Y$

This step of adding a self-model to a base network is also called *network reification*. If such self-model states are dynamic, they describe adaptive network characteristics. In a graphical 3D-format, such self-model states are depicted at a next level (also called *reification level*), where the original network is at a *base level*. As an example, the weight $\boldsymbol{\omega}_{X,Y}$ of a connection from state X to state Y can be represented (at a next reification level) by a self-model state named $\mathbf{W}_{X,Y}$ (e.g., for an objective representation) or $\mathbf{RW}_{X,Y}$ (e.g., for a subjective representation).

Having self-model states to model an adaptation principle in a network-oriented manner is only a first step. To fully model a certain adaptation principle by a self-modeling network, the dynamics of each self-model state itself and its effect on a corresponding target state Y have to be specified in a network-oriented manner by the three general standard types of network structure characteristics a) *connectivity*, b) *aggregation*, and c) *timing*:

Connectivity for the self-model states in a self-modeling network

For the self-model states, their *connectivity* in terms of their incoming and outgoing connections has two different functions:

Effectuating its special effect from its specific role

The *outgoing downward causal connections* from the self-model states $\mathbf{W}_{X,Y}$, $\mathbf{C}_{j,Y}$, $\mathbf{P}_{i,j,Y}$, $\mathbf{H}_Y$ to state Y represent the specific causal impact (their special effect from their specific role) each of these self-model states has on Y. These downward causal impacts are standard per role, and make that the adaptive values $\mathbf{W}_{X,Y}(t)$, $\mathbf{C}_{j,Y}(t)$, $\mathbf{P}_{i,j,Y}(t)$, $\mathbf{H}_Y(t)$ at t are actually used for the adaptive characteristics of the base network in equations (1) and (2).

Indicating the input for the adaptation principle as specified in b)

The *incoming upward or leveled connections* to a self-model state are used to specify the *input* needed for the particular adaptation principle that is addressed.

Aggregation for the self-model states in a self-modeling network

For the self-model states, their aggregation characteristics have one main aim:

Expressing the adaptation principle by a mathematical function

For the *aggregation* of the incoming causal impacts for a self-model state, provided as indicated in a), a specific combination function is chosen *to express the adaptation principle* in a declarative mathematical manner.

Timing for the self-model states in a self-modeling network

For the self-model states, their timing characteristics have one main aim:

Expressing the adaptation speed for the adaptation principle by a number

Finally, like any other state, self-model states have their own *timing* in terms of speed factors. These speed

factors are used as the means to express the adaptation speed.

As a base network extended by including a self-model is also a temporal-causal network model itself, as has been shown in [7], Ch 10, this self-modeling construction can easily be applied iteratively to include self-models of multiple (reification) levels. This can provide higher-order adaptive network models, and has turned out quite useful to model, for example, within Cognitive Neuroscience plasticity and metaplasticity (e.g., [20-23]) in a unified form by a second-order adaptive mental causal network with three levels, one base level and a first- and a second-order self-model level for causation concerning plasticity and metaplasticity, respectively, as shown in [7], Ch 4.

In the current paper, the notion of a multi-level self-modeling network will be illustrated by a higher-order adaptive social network model. In this model, in addition to the Euclidean combination function described in Section 2.1, two other combination functions from the library are used:

the advanced logistic sum combination function $\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k)$ defined by:

$$\mathbf{alogistic}_{\sigma,\tau_{\mathbf{log}}}(V_1, \ldots, V_k) = [\frac{1}{1+e^{-\sigma(V_1+\cdots+V_k-\tau_{\mathbf{log}})}} - \frac{1}{1+e^{\sigma\tau_{\mathbf{log}}}}](1+e^{-\sigma\tau_{\mathbf{log}}}) \quad (4)$$

where σ is a steepness parameter and $\tau_{\mathbf{log}}$ a threshold parameter and V_1, …, V_k are the impacts from the states from which the considered state Y gets incoming connections

the simple linear homophily combination function $\mathbf{slhomo}_{\alpha,\tau_{\mathbf{hom}}}(V_1, V_2, W)$ defined by

$$\mathbf{slhomo}_{\alpha,\tau_{\mathbf{hom}}}(V_1, V_2, W) = W + \alpha\, W\,(1-W)\,(\tau_{\mathbf{homo}} - |V_1 - V_2|) \quad (5)$$

where α is an amplification parameter and $\tau_{\mathbf{hom}}$ a tipping point parameter and V_1, V_2 are a person's representations of the two persons' states involved and W represents the weight of their connection

Here, $\mathbf{slhomo}_{\alpha,\tau_{\mathbf{hom}}}(V_1, V_2, W)$ is used to model bonding based on (faked) homophily by internal connection weight representations, and $\mathbf{alogistic}_{\sigma,\tau_{\mathbf{log}}}(\ldots)$ to model control of the bonding. Bonding based on homophily [17-19] is the social network adaptation principle that is sometimes expressed by

'Birds of a feather flock together'

This expresses how being 'birds of a feather' or 'being alike' (modeled by state values V_1 and V_2 for the two persons not differing much) causally affects the connection between two persons. Note that the homophily tipping point $\tau_{\mathbf{hom}}$ is the point where the difference between the states of the two individuals (represented by $|V_1 - V_2|$) turns an increase of bonding (outcome > W) into a decrease (outcome < W), and conversely. In Section 4 this tipping point is set at 0.25: so in that case a difference $|V_1 - V_2| < 0.25$ has as causal effect that the connection will be strengthened (increase of W), whereas a difference $|V_1 - V_2| > 0.25$ has as causal effect that it will be weakened (decrease of W).

This shows an example of how for a social application domain, within a causal network, states can have a causal effect on network connections. By applying a self-modeling network model, this form of causation (for adaptation of connections through bonding by homophily) together with the causation between states in the base network (for social contagion) is addressed in a unified manner by one overall (two-level) causal network model, in contrast to the commonly used hybrid modeling approach to adaptive networks pointed out above in the second paragraph of this Section 2.2. Moreover, in Section 3 it will be shown how also a third level for the control of the adaptation process can be incorporated within such a self-modeling causal network, thus obtaining a three-level network model unifying within one causal model the base network dynamics with adaptation of the connections of the base network and the control of that adaptation.

III. A Social Causal Network with Controlled Adaptation

To illustrate the use of self-modeling networks to incorporate in a unified manner both dynamics and (multi-order) adaptivity in a causal model, this section presents an adaptive causal network model for controlled bonding based on homophily by using subjective representations (some of which are based on fake input). The presented causal network model integrates three types of interacting processes, modeled within the causal model at three different levels:

The considered social base network itself with its (within-network) dynamics for social contagion [24]

Change of this social network over time based on bonding by homophily [17-19]: first-order social network adaptation

Control of the first-order social network adaptation: second-order social network adaptation

In contrast to what is usually done, for example, also in [19], here the bonding is not assumed to depend on the objective states for the two persons, but on how these states are perceived and represented by the persons through the formation of subjective state representation states. By controlling the formation of these subjective state representation states, indirectly the bonding is affected; contrarely, if you don't take care to acquire information about the other person, then you miss a good reason for stronger or weaker bonding. To cover this, the above three types of processes have been modeled by a second-order adaptive causal network model based on a multi-level self-modeling network using a first-order self-model (for formation of the subjective state representation states and for the bonding based on them) and a second-order self-model (for the control of the formation of the subjective representation states). That offers some room to model cheating about one's own properties, as regularly happens in real life: by faking an own state, the other person will make a false representation for it, which then will affect that person's bonding in a false manner.

The model's connectivity is depicted in Fig. 1 by an example for two persons, one of which is faking his or her properties in order to achieve successful bonding. In this 3D picture, each of the three planes models one of the

three types of processes mentioned above; for an explanation of the states, see Table 1.

Table 1 Types of states in the introduced controlled adaptive social network model

$\mathbf{S}_A$	Objective state *Z* of person A
$\mathbf{S}_B$	Objective state *Z* of person B
$\mathbf{FS}_B$	Objective state of person B faking state *Z* of person A
$\mathbf{RS}_{A,A}$	Subjective representation of person A for state *Z* of person A
$\mathbf{RS}_{B,B}$	Subjective representation of person B for state *Z* of person B
$\mathbf{RS}_{A,B}$	Subjective representation of person B for state *Z* of person A
$\mathbf{RS}_{B,A}$	Subjective representation of person A for state *Z* of person B
$\mathbf{RFS}_{B,B}$	Subjective representation of person B for his or her faked state *Z*
$\mathbf{RW}_{A,B}$	Subjective representation of person A for the weight of the connection from person A to person B
$\mathbf{RW}_{B,A}$	Subjective representation of person B for the weight of the connection from person B to person A
$\mathbf{CC}_{A,B}$	Control state for communication from A to B for the state *Z* of A: representation of the weight of the connection from $\mathbf{RS}_{A,A}$ to $\mathbf{RS}_{A,B}$
$\mathbf{CC}_{B,A}$	Control state for communication from B to A for the state *Z* of B: representation of the weight of the connection from $\mathbf{RS}_{B,B}$ to $\mathbf{RS}_{B,A}$
$\mathbf{CO}_{A,B}$	Control state for observation by B for the state *Z* of A observed by B: representation of the weight of the connection from $\mathbf{S}_A$ to $\mathbf{RS}_{A,B}$
$\mathbf{CO}_{B,A}$	Control state for observation by A for the state *Z* of B observed by A: representation of the weight of the connection from $\mathbf{S}_B$ to $\mathbf{RS}_{B,A}$

The types of connections used at and between the three levels within this network model are shown in Table 2. Here *Z* is a type of state of a person, for example, how often the person listens to a certain type of music; to keep the notations simple, this type is left out of them; if needed, the *Z* could be used as an additional subscript.

At the base level, social contagion is modelled by intralevel connections (depicted by black arrows in the lower plane in Fig. 1) such as $\mathbf{S}_A \rightarrow \mathbf{S}_B$, $\mathbf{FS}_B \rightarrow \mathbf{S}_A$, and $\mathbf{S}_A \rightarrow \mathbf{FS}_B$. Here the last connection models B faking by intentionally listening to the same type of music as A just at the moments that A can observe it. In contrast to $\mathbf{FS}_B$, state $\mathbf{S}_B$ indicates how much B normally listens to that type of music. In the simulated scenario, $\mathbf{S}_A$ will have high values and $\mathbf{S}_B$ low values, whereas by copying $\mathbf{S}_A$ also $\mathbf{FS}_B$ gets high values.

Within the first-order self-model, each person has subjective internal representation states of other persons' states *Z* and the of state *Z* of her or himself, and also of his or her connections to others. This first-order self-model is modeled in the middle plane. For example, person A's internal representation state for person B having state *Z* is modeled by state representation $\mathbf{RS}_{B,A}$, and A's subjective representation of his or her connection to B is modeled by connection weight representation $\mathbf{RW}_{A,B}$.

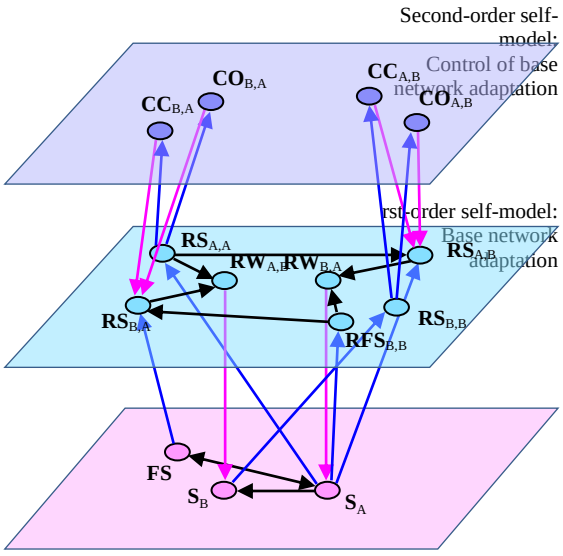


Figure 1 Overview of the connectivity of the second-order adaptive social network model for bonding by homophily for two persons A and B, where B is faking the homophily for A.

There are two pathways that contribute to formation of state representations such as $\mathbf{RS}_{A,B}$. First, these representations can be obtained through observation of $\mathbf{S}_A$ by B. This is modeled by an upward interlevel connection $\mathbf{S}_A \rightarrow \mathbf{RS}_{A,B}$ from the base network to the first-order self-model. As B is faking his or her base state, observation by A is modeled *not* by a connection $\mathbf{S}_B \rightarrow \mathbf{RS}_{B,A}$ but by connection $\mathbf{FS}_B \rightarrow \mathbf{RS}_{B,A}$.

A second pathway for a person B to get information on person A's state is through communication between persons. For example, if A communicates his or her subjective representation $\mathbf{RS}_{A,A}$ of the own state $\mathbf{S}_A$ to B (e.g., 'I often play this type of music!'), this is modeled by an intralevel connection $\mathbf{RS}_{A,A} \rightarrow \mathbf{RS}_{A,B}$ within the middle plane for the first-order self-model. Also in the communication, B is faking; therefore communication from B to A is *not* modeled by a connection $\mathbf{RS}_{A,B} \rightarrow \mathbf{RS}_{B,A}$, but by connection $\mathbf{RFS}_{B,B} \rightarrow \mathbf{RS}_{B,A}$ (so that B may falsely communicate 'What a coincidence, I also often play that type of music!').

Table 2 Connections in the controlled adaptive social network model and their explanation

Intralevel connections	
$\mathbf{S}_A \rightarrow \mathbf{S}_B$	Social contagion from A to B for state *Z*
$\mathbf{FS}_B \rightarrow \mathbf{S}_A$	Social contagion from B's faked state for *Z* to A
$\mathbf{S}_A \rightarrow \mathbf{FS}_B$	Faking contagion from state *Z* of A to faked state *Z* of B
$\mathbf{RS}_{A,A} \rightarrow \mathbf{RS}_{A,B}$	Communication of state *Z* from A to B
$\mathbf{RFS}_{B,B} \rightarrow \mathbf{RS}_{B,A}$	Communication of faked state *Z* from B to A
$\mathbf{RS}_{A,A} \rightarrow \mathbf{RW}_{A,B}$	Effect of represented state *Z* of A by A on the connection from A to B (bonding by homophily)
$\mathbf{RS}_{B,A} \rightarrow \mathbf{RW}_{A,B}$	Effect of represented state *Z* of B by A on the connection from A to B (bonding by homophily)
$\mathbf{RFS}_{B,B} \rightarrow \mathbf{RW}_{B,A}$	Effect of represented faked state *Z* of B by B on the connection from B to A (bonding by homophily)
$\mathbf{RS}_{A,B} \rightarrow \mathbf{RW}_{B,A}$	Effect of represented state *Z* of A by B on the connection from B to A (bonding by homophily)
Interlevel connections	

$\mathbf{S}_A \rightarrow \mathbf{RS}_{A,A}$	Impact of observation of A's state *Z* by A on A's representation of A's state *Z*	Upward from base network to first-order self-model
$\mathbf{S}_B \rightarrow \mathbf{RS}_{B,B}$	Impact of observation of B's state *Z* by B on B's representation of B's state *Z*	
$\mathbf{S}_A \rightarrow \mathbf{RS}_{A,B}$	Impact of observation of A's state *Z* by B on B's representation of A's state *Z*	
$\mathbf{FS}_B \rightarrow \mathbf{RS}_{B,A}$	Impact of observation of B's faked state *Z* by A on A's representation of B's state *Z*	
$\mathbf{RW}_{A,B} \rightarrow \mathbf{S}_B$	Effectuation of base connection weight for social contagion from state *Z* of A to state *Z* of B	Downward from first-order self-model to base network
$\mathbf{RW}_{B,A} \rightarrow \mathbf{S}_A$	Effectuation of base connection weight for social contagion from faked state *Z* of B to state *Z* of A	
$\mathbf{RS}_{A,A} \rightarrow \mathbf{CC}_{B,A}$	Communication control monitoring connection for A	Upward from first-order self-model to second-order self-model
$\mathbf{RS}_{B,B} \rightarrow \mathbf{CC}_{A,B}$	Communication control monitoring connection for B	
$\mathbf{RS}_{A,A} \rightarrow \mathbf{CO}_{B,A}$	Observation control monitoring connection for A	

$\mathbf{RW}_{A,B}$ and $\mathbf{RS}_{B,A} \rightarrow \mathbf{RW}_{A,B}$ within the first-order self-model. The connection representations by **RW**-states in turn affect the social contagion within the social network, which is modeled by downward interlevel connections $\mathbf{RW}_{A,B} \rightarrow \mathbf{S}_B$ and $\mathbf{RW}_{B,A} \rightarrow \mathbf{S}_A$ from the first-order self-model in the middle plane to the base network.

To control the social network adaptation processes, two types of control actions are considered in particular:

controlling the observation of state *Z* from person A by person B is modeled by control state $\mathbf{CO}_{A,B}$ and from person B by person A is modeled by control state $\mathbf{CO}_{B,A}$

controlling the communication about state *Z* from person A to person B, modeled by control state $\mathbf{CC}_{A,B}$ and the communication about state *Z* from person B to person A, is modeled by control state $\mathbf{CC}_{B,A}$

Activation of a communication control state makes that the related connection in the first-order self-model in the middle plane gets a high value (1 or close to 1); this is achieved by interlevel connections from control states to **RS**-states in the first-order self-model. For example, activation of communication control state $\mathbf{CC}_{A,B}$ makes that the connection $\mathbf{RS}_{A,A} \rightarrow \mathbf{RS}_{A,B}$ from A's state $\mathbf{RS}_{A,A}$ to B's state $\mathbf{RS}_{A,B}$ gets a high value (1 or close to 1) so that

mb base connectivity

		1	2	3
X_1	$\mathbf{S}_A$	X_3		
X_2	$\mathbf{S}_B$	X_1		
X_3	$\mathbf{FS}_B$	X_1		
X_4	$\mathbf{RS}_{A,A}$	X_1		
X_5	$\mathbf{RS}_{B,B}$	X_2		
X_6	$\mathbf{RFS}_{B,B}$	X_1		
X_7	$\mathbf{RS}_{A,B}$	X_1	X_4	
X_8	$\mathbf{RS}_{B,A}$	X_3	X_6	
X_9	$\mathbf{RW}_{A,B}$	X_4	X_8	X_9
X_{10}	$\mathbf{RW}_{B,A}$	X_6	X_7	X_{10}
X_{11}	$\mathbf{CC}_{A,B}$	X_4		
X_{12}	$\mathbf{CC}_{B,A}$	X_3		
X_{13}	$\mathbf{CO}_{A,B}$	X_4		
X_{14}	$\mathbf{CO}_{B,A}$	X_3		

mcw connection weights

		1	2	3
X_1	$\mathbf{S}_A$	X_{10}		
X_2	$\mathbf{S}_B$	X_9		
X_3	$\mathbf{FS}_B$	1		
X_4	$\mathbf{RS}_{A,A}$	1		
X_5	$\mathbf{RS}_{B,B}$	1		
X_6	$\mathbf{RFS}_{B,B}$	1		
X_7	$\mathbf{RS}_{A,B}$	X_{13}	X_{11}	
X_8	$\mathbf{RS}_{B,A}$	X_{14}	X_{12}	
X_9	$\mathbf{RW}_{A,B}$	1	1	1
X_{10}	$\mathbf{RW}_{B,A}$	1	1	1
X_{11}	$\mathbf{CC}_{A,B}$	1		
X_{12}	$\mathbf{CC}_{B,A}$	1		
X_{13}	$\mathbf{CO}_{A,B}$	1		
X_{14}	$\mathbf{CO}_{B,A}$	1		

ms speed factors

		1
X_1	$\mathbf{S}_A$	0.0005
X_2	$\mathbf{S}_B$	0.0005
X_3	$\mathbf{FS}_B$	0.8
X_4	$\mathbf{RS}_{A,A}$	0.9
X_5	$\mathbf{RS}_{B,B}$	0.9
X_6	$\mathbf{RFS}_{B,B}$	0.9
X_7	$\mathbf{RS}_{A,B}$	0.9
X_8	$\mathbf{RS}_{B,A}$	0.9
X_9	$\mathbf{RW}_{A,B}$	0.1
X_{10}	$\mathbf{RW}_{B,A}$	0.1
X_{11}	$\mathbf{CC}_{A,B}$	0.2
X_{12}	$\mathbf{CC}_{B,A}$	0.2
X_{13}	$\mathbf{CO}_{A,B}$	0.4
X_{14}	$\mathbf{CO}_{B,A}$	0.4

mcfw combination function weights

		1 eucl	2 slhomo	3 alogistic
X_1	$\mathbf{S}_A$	1		
X_2	$\mathbf{S}_B$	1		
X_3	$\mathbf{FS}_B$	1		
X_4	$\mathbf{RS}_{A,A}$	1		
X_5	$\mathbf{RS}_{B,B}$	1		
X_6	$\mathbf{RFS}_{B,B}$	1		
X_7	$\mathbf{RS}_{A,B}$	1		
X_8	$\mathbf{RS}_{B,A}$	1		
X_9	$\mathbf{RW}_{A,B}$		1	
X_{10}	$\mathbf{RW}_{B,A}$		1	
X_{11}	$\mathbf{CC}_{A,B}$			1
X_{12}	$\mathbf{CC}_{B,A}$			1
X_{13}	$\mathbf{CO}_{A,B}$			1
X_{14}	$\mathbf{CO}_{B,A}$			1

mcfp combination function parameters

		1 eucl		2 slhomo		3 alogistic	
		1	2	1	2	1	2
		n	λ	α	τ_{hom}	σ	τ_{log}
X_1	$\mathbf{S}_A$	1	1				
X_2	$\mathbf{S}_B$	1	1				
X_3	$\mathbf{FS}_B$	1	1				
X_4	$\mathbf{RS}_{A,A}$	1	1				
X_5	$\mathbf{RS}_{B,B}$	1	1				
X_6	$\mathbf{RFS}_{B,B}$	1	1				
X_7	$\mathbf{RS}_{A,B}$	1	2				
X_8	$\mathbf{RS}_{B,A}$	1	2				
X_9	$\mathbf{RW}_{A,B}$			3	0.25		
X_{10}	$\mathbf{RW}_{B,A}$			3	0.25		
X_{11}	$\mathbf{CC}_{A,B}$					50	0.1
X_{12}	$\mathbf{CC}_{B,A}$					50	0.1
X_{13}	$\mathbf{CO}_{A,B}$					50	0.1
X_{14}	$\mathbf{CO}_{B,A}$					50	0.1

iv initial values

		1
X_1	$\mathbf{S}_A$	0.9
X_2	$\mathbf{S}_B$	0.2
X_3	$\mathbf{FS}_B$	0.2
X_4	$\mathbf{RS}_{A,A}$	0.7
X_5	$\mathbf{RS}_{B,B}$	0.4
X_6	$\mathbf{RFS}_{B,B}$	0.5
X_7	$\mathbf{RS}_{A,B}$	0.5
X_8	$\mathbf{RS}_{B,A}$	0.5
X_9	$\mathbf{RW}_{A,B}$	0.5
X_{10}	$\mathbf{RW}_{B,A}$	0.5
X_{11}	$\mathbf{CC}_{A,B}$	0
X_{12}	$\mathbf{CC}_{B,A}$	0
X_{13}	$\mathbf{CO}_{A,B}$	0
X_{14}	$\mathbf{CO}_{B,A}$	0

Box 1

Full specification of the adaptive self-modeling causal network model by role matrices for all (connectivity, aggregation and timing) characteristics causally affecting the network states

As indicated, person A's representation of her or his connection to person B is modeled by $\mathbf{RW}_{A,B}$. It is assumed that for the bonding by homophily adaptation principle, the adaptive change of the represented connection for A to B depends on the internal representation states $\mathbf{RS}_{B,A}$ and $\mathbf{RS}_{A,A}$. Therefore, this adaptation is supported by intralevel connections $\mathbf{RS}_{A,A} \rightarrow$ the transfer of information by communication happens; this is modeled by interlevel connection $\mathbf{CO}_{A,B} \rightarrow \mathbf{RS}_{A,B}$. This can be considered as B asking A for the information about him or herself, upon which A communicates this information. Similarly, activation of an observation control state $\mathbf{CO}_{A,B}$ makes that the connection $\mathbf{S}_A \rightarrow \mathbf{RS}_{A,B}$ from A's state $\mathbf{S}_A$ to B's state $\mathbf{RS}_{A,B}$ gets a high value (1 or close

to 1) so that the transfer of information by observation takes place; this is modeled by connection $\mathbf{CO}_{A,B} \rightarrow \mathbf{RS}_{A,B}$. In the case modeled here, control states such as $\mathbf{CC}_{A,B}$ and $\mathbf{CO}_{A,B}$ themselves may become active depending on B's state $\mathbf{RS}_{B,B}$; this is modeled by connections $\mathbf{RS}_{B,B} \rightarrow \mathbf{CC}_{A,B}$ and $\mathbf{RS}_{B,B} \rightarrow \mathbf{CO}_{A,B}$. But this may be addressed in many other ways as well, including externally determined control, for example, by enabling or allowing observation or communication (only) at specific time slots.

To specify a network model according to the approach described in [7], as discussed in Section 2, three types of network characteristics are to be covered: *connectivity*, *aggregation* and *timing* characteristics. Any state in the network is causally affected by all of such characteristics, each from its own specific role. Following the role matrices specification format defined in [7] (pp. 39-41, 89), they are specified by role matrices as shown in Box 1 which are used as input for the dedicated software environment to automatically obtain the simulation discussed In Section 4.

More specifically, *role matrices* indicate in rows successively for all network states, the factors that causally affect them from the different roles. So in the row for a state *Y*, in each column a causal relation is specified affecting state *Y* for the role described by that role matrix. In this way, role matrices describe the network model by mathematical relations and functions.

In the first place, concerning *connectivity* roles, each state is causally affected by the other states from which it has incoming connections and by the weights of these connections. In role matrix **mb** (see Box 1), for each state it is indicated from which other states it has incoming connections from the same or a lower level. In role matrix **mcw**, it is indicated what are the connection weights for the connected states indicated in **mb**. If these weights are static, their value is indicated, in green shaded cells (here always 1), but if the connection weight is adaptive, instead of a number the self-model state representing this weight is indicated in role matrix **mcw**. This can be seen (cells shaded in a peach-red colour) in **mcw** for the incoming connections for the first two states X_1 and X_2, and for the incoming connections for the states X_7 and X_8. Indicating these adaptive value representations, defines the downward connections of Fig. 1. From the *timing role*, also its speed factor causally affects a state; they are shown in Box 1 (role matrix **ms**, which actually is a vector).

In the lower part of Box 1, showing the *aggregation* roles causally affecting a state, it can be seen which states use which combination functions (role matrix **mcfw**) and which parameter values for them (role matrix **mcfp**). In addition to the five role matrices for the different roles of causal impacts, the initial values for the example simulation are also shown in Box 1, which may be considered as initial causal impacts.

IV. Simulation: Faking Homophily for Bonding

In this section, a simulation of a simulated example scenario will be discussed to illustrate the introduced second-order adaptive causal social network model for faking homophily. In Fig. 2 the simulation for the example scenario is shown. Here the states $\mathbf{S}_X$ are slowly changing whereas the connection representations in the form of the **RW**-states are changing faster. It indeed can be seen that for A and B both directional connection representations $\mathbf{RW}_{A,B}$ and $\mathbf{RW}_{B,A}$ start to gradually increase from time point 5 on to reach values above 0.7 which in the long run eventually reach a value (close to) 1. These changes of the connections are a consequence of the homophily principle, as the values of state $\mathbf{S}_A$ of A and the faked states $\mathbf{FS}_B$ and $\mathbf{RFS}_{B,B}$ for B quickly get close to each other; note that the tipping point for similarity set was 0.25, so a difference between the relevant representation states < 0.25 is strengthening a connection.

In Fig. 2, also the roles that are played by the control states in the form of the **CO**- and **CC**-states and by the **RS**-states for subjective representations can be seen. The two lines that start at 0 and get close to 1 around or soon after time 10 indicate the control states $\mathbf{CO}_{A,B}$ and $\mathbf{CO}_{B,A}$ (light green) for observation and $\mathbf{CC}_{A,B}$ and $\mathbf{CC}_{B,A}$ (light blue) for communication, respectively. This makes that at that time their mutual observation and communication channels $\mathbf{S}_A \rightarrow \mathbf{RS}_{A,B}$ and $\mathbf{FS}_B \rightarrow \mathbf{RS}_{B,A}$, and $\mathbf{RS}_{A,A} \rightarrow \mathbf{RS}_{A,B}$ and $\mathbf{RFS}_{B,B} \rightarrow \mathbf{RS}_{B,A}$ get weights close to 1. This implies that then they indeed both observe and communicate to each other about the type of music they usually listen to. These control states are triggered in this example scenario because each of the persons automatically observes his or herself and therefore they quickly (before time point 4) form representation states $\mathbf{RS}_{A,A}$ and $\mathbf{RS}_{B,B}$ of their own **S**-states concerning music (the red lines, starting at 0.4 for B and at 0.7 for A).

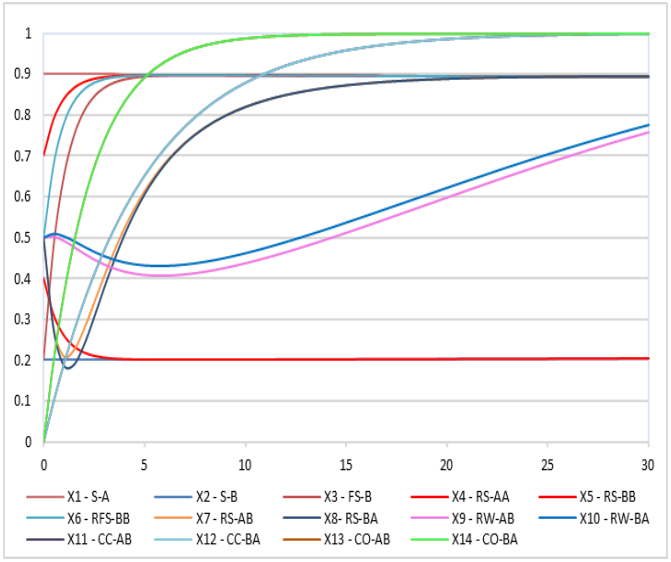


Figure 2 Outcomes for the example scenario simulation

Because of these communication and observation actions, the mutual subjective representations $\mathbf{RS}_{A,B}$ of B about A (the dark green line) and $\mathbf{RS}_{B,A}$ of A about B (the orange line) based on fake information are formed, and around time 20 reach levels close to 0.9. Only now these subjective representations have been formed in a controlled manner, the homophily principle can start to work, as the bonding works through the (subjective) representation **RS**-states, not through the (objective) states $\mathbf{S}_X$ themselves. More specifically, from the moment on that the subjective representations of A about B and A's own subjective representation about her- or himself get closer

than 0.25 (which is just before time point 5), her/his self-model representation $\mathbf{RW}_{A,B}$ of her connection to B (the pink line) starts to gradually increase. Similarly, the effect of the subjective representations of B for A and B's own subjective self-model representation about him or herself, on the subsequent increase of his representation $\mathbf{RW}_{B,A}$ of this connection to A (the blue line) can be noted. Before that point in time their connections were not increasing, but instead go slightly downward; this illustrates the effect of the control via the subjective self-model representation states on the adaptation.

V. Discussion

Causal modeling combines two quite useful properties. In the first place, it is an intuitive, declarative way of modeling supported by often used graphical representations. Secondly, due to the universal character of causality, in principle it should apply to practically all scientific disciplines. However, limitations for dynamics and adaptivity stand in the way of applicability in many domains. In this paper it was discussed how these challenges can be addressed by exploiting the notion of self-modeling network developed from a Network Science perspective [6, 7]. Self-modeling causal networks cover dynamics of the states of the nodes as well as adaptivity of these causal relations. Here adaptivity of a base network is obtained by explicit representations of the characteristics of the causal relations in the form of a self-model added to the base network. These self-modeling causal networks are specified in a declarative manner by mathematical relations and functions, and provide a causal network addressing the adaptation. Therefore, this approach indeed takes causal modeling to a next level so that now dynamics and adaptivity are also covered by a unified causal perspective. By an illustration for a controlled adaptive social causal network model, it has been shown how this widens the scope of applicability of causal modeling.

Another topic that illustrates the applicability of the causal modeling approach based on self-modeling networks well is plasticity and metaplasticity within Cognitive Neuroscience, as described, for example, in empirical literature such as [20-23]. In [7], Ch 4, it is shown how this can be modeled as a self-modeling causal network incorporating a first-order self-model for plasticity and a second-order self-model for metaplasticity.

Such multi-level self-modeling causal networks incorporate different types of causation. In the first place this covers causation between base states, as is a familiar form of causation known from traditional causal models. This is also the form of causation usually focused on (for mental states) within Philosophy of Mind, such as in [14]. However, in self-modeling network models there is also causation from these base states to other types of states representing causal relations, and back. Such forms of causation have to occur as soon as causal relations can change in the world, as such change should be caused by something. In turn, such changes causally affect the future processes.

So, for adaptive cases, from a completeness of causation perspective such less familiar forms of causation cannot be avoided, and have direct relations to what actually happens in the world. Indeed, for example in empirically focussed Cognitive Neuroscience literature such as [20-23], it is described in some detail how states and processes addressing plasticity and metaplasticity are realised by specific (changing) brain configurations and causal relations for them. So, self-models are not just artificial modeling concepts created by some fantasy: they relate to real counterparts of them in the physical world. In that sense, it may be claimed that self-modeling causal networks actually exist in the world, at least for this context of Cognitive Neuroscience. A similar illustration for the biological domain can be found in [7], Ch. 7, addressing a five-level self-modeling causal network model describing different stages in an evolutionary process. Here the different types of states and causation in the self-modeling causal network have counterparts in the physical world in the form of (changing) configurations and processes as described in literature from Biology.

The presented approach allows declarative modeling of dynamic and adaptive behaviour of multiple orders of adaptation from a unified causal perspective. Traditionally, declarative modeling approaches are a strong focus of AI. There are two longstanding themes in AI to which the work presented here relates in particular: causal modeling as already mentioned [1-4] and metalevel architectures and metaprogramming [9-13]. As discussed, a main contribution to the causal modeling area is that this is extended by dynamics and adaptivity of the causal modeling, addressing both the dynamics of the causal effects and the adaptive dynamics of the causal relations themselves. A main contribution to the area of metalevel architectures and metaprogramming is that now network models are covered as well in the form of self-modeling networks, while traditionally the focus in this area is mainly on logical, functional and object-oriented modeling or programming approaches; e.g., [10].

In relation to the area of Neural Networks within AI, the network-oriented modeling approach described here distinguishes itself by a multidisciplinary Network Science focus on causality and adaptation within empirical natural and human-directed sciences. In contrast, the area of Neural Networks has its main focus on artificial neural networks to solve optimisation challenges and on their computational efficiency. Another important distinction is the notion of self-modeling network which is the main focus in the current paper. However, there are also some technical elements in common, for example, the format of the canonical difference equation (1) (see Section 2.1) used here can be considered a form of socalled recurrent network as also used in the Neural Networks area. But a difference here is the use of speed factors per node which enables to model different nodes that are not necessarily synchronous in their dynamics. This asynchrony is usually needed to model real-world processes as these are not often synchronous and can even involve entirely different time scales. This explicit way to model differentiated timing is not a common practice in the Neural Networks area within AI.

From a more theoretical side, following Ashby [26] in [25] Section 3.1 it has been shown that any state-

determined dynamical system (as defined in [26] and also used in [27]) can be described by a set of first-order differential equations, and conversely. Moreover, in [25], Section 3.2 it has also been shown how any set of first-order differential equations can be (re)modeled by a temporal-causal network model. It has been shown in [7], Ch 10 that any self-modeling network obtained by adding a self-model to a temporal-causal network is iself also a temporal-causal network. Therefore, these methods can also be applied to adaptive processes: any description of an adaptation process by a state-determined system or by first-order differential equations can be rewritten as a self-model in temporal-causal network format. This provides evidence from a more theoretical analysis perspective that the approach discussed here has a wide scope of applicability.

There are still some more interesting challenges that can be addressed. A first challenge is to explore other interesting cases of higher-order adaptation and to investigate whether self-modeling causal networks indeed are suitable to model them. Within Cognitive Neuroscience, from an empirical perspective the notions of plasticity and metaplasticity have been introduced [20-23], relating to first- and second-order adaptation. It has been found how these can be modeled by a second-order self-modeling network; see [6] and [7], Ch 4. Similarly, it has been described how second-order adaptive social networks for bonding by homophily can be modeled by self-modeling networks; see [6] and [7], Ch 6. However, in general higher-order adaptation for social networks has not been addressed well in the literature. As an exception, in [28, 29] the notion of inhibiting adaptation for networks has been described, which refers to some form of second-order adaptive social networks. This applies, for example, to terrorist network organisations. It would be interesting to investigate whether and how such second-order social networks can also be described as self-modeling causal networks.

Within Biology, some literature can be found on how evolutionary processes can be described as higher-order adaptation; e.g., [30, 31]. It has been shown in [7], Ch 7, how one case study concerning pregnancy and disgust can be modeled by a fourth-order adaptive self-modeling causal network model. It is interesting to address more case studies in this area. Moreover, in Hofstadter [32] claims that the notion of Strange Loop underlies human intelligence. This is described in [32] informally as a form of self-modeling of multiple levels, where for some n, the n^{th} level is equal to the base level, so that the levels form a cycle. It has been found that this also can be modeled by a self-modeling network; see [7], Ch 8 for an example for a mental network and [33] for an example for a social network. However, the notion of Strange Loop could be explored for more cases.

Finally, as mentioned the research described in the current paper follows the multidisciplinary perspective of Network Science. Therefore, the focus is on adaptation principles known from nature and described in empirical disciplines such as Biology, Neuroscience, Cognitive Science or Social Sciences. In contrast, it may be an interesting challenge to investigate how some wellknown artificial methods for machine learning can be modeled by self-modeling networks. As the self-modeling network approach provides a declarative perspective on modeling adaptation processes, this might provide more declarative descriptions of such artifical methods, which usually are described in a procedural manner by algorithms. As pointed out in one of the paragraphs above, from a theoretical perspective this should be possible. But it would be interesting to see how this would actually look like for some examples. It might provide a more clear modeling separation of the conceptual core of a machine learning method and the procedural optimisation involved. As an example, in this way backpropagation for artificial neural networks could be modeled in a network-oriented manner with gradient descent as conceptual core plus an efficient procedure to do the required calculations; e.g., [34], Ch 7.

REFERENCES

[1] Kuipers, B.J.: Commonsense reasoning about causality: Deriving behavior from structure, Artificial Intelligence 24**,** 169--203 (1984)

[2] Kuipers, B.J., Kassirer, J.P.: How to discover a knowledge representation for causal reasoning by studying an expert physician. In: Proc. of the Eighth International Joint Conference on Artificial Intelligence, IJCAI'83, pp. 49--56. William Kaufman, Los Altos, CA, (1983)

[3] Pearl, J.: Causality. Cambridge University Press (2000)

[4] Wright, S.: Correlation and Causation. Journal of Agricultural Research 20, 557--585 (1921)

[5] Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer Publishers (2016)

[6] Treur, J.: Modeling Higher-Order Adaptivity of a Network by Multilevel Network Reification. Network Science 8, S110--S144 (2020)

[7] Treur, J.: Network-oriented modeling for adaptive networks: Designing Higher-order Adaptive Biological, Mental and Social Network Models. Cham, Switzerland: Springer Nature Publishing (2020)

[8] Treur, J.: Multilevel Network Reification: Representing Higher Order Adaptivity in a Network. In: Aiello, L., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L. (eds), Proc. of the 7th Int. Conf. on Complex Networks and their Applications, ComplexNetworks'18, vol. 1. Studies in Computational Intelligence, vol. 812, pp. 635--651, Springer Nature (2018)

[9] Bowen, K.A., and Kowalski, R.: Amalgamating language and meta-language in logic programming. In Logic Programming, K. Clark and S. Tarnlund, Eds. Academic Press, New York, pp. 153--172 (1982)

[10] Demers, F.N., Malenfant, J.: Reflection in logic, functional and objectoriented programming: a Short Comparative Study. In IJCAI'95 Workshop on Reflection and Meta-Level Architecture and their Application in AI, pp. 29--38 (1995)

[11] Sterling, L., Shapiro, E.: The Art of Prolog. MIT Press (Ch 17, pp. 319--356) (1996)

[12] Sterling, L., Beer, R.: Metainterpreters for expert system construction. Journal of Logic Programming 6, 163--178 (1989)

[13] Weyhrauch, R.W.: Prolegomena to a Theory of Mechanized Formal Reasoning. Artificial Intelligence 13, 133–170 (1980)

[14] Kim, J. (1996). Philosophy of Mind. Westview Press (1996)
[15] Mooij, J.M., Janzing, D., Schölkopf, B.: From Differential Equations to Structural Causal Models: the Deterministic Case. In: Nicholson, A., and Smyth, P. (eds.), Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI-13), pp. 440--448. AUAI Press (2013)
[16] Hebb, D.O.: The organization of behavior: A neuropsychological theory. Wiley (1949)
[20] Abraham, W.C., Bear, M.F.: Metaplasticity: the plasticity of synaptic plasticity. Trends in Neuroscience 19(4), 126--130 (1996)
[21] Garcia, R.: Stress, Metaplasticity, and Antidepressants. Current Molecular Medicine 2, 629--638 (2002)
[22] Magerl, W., Hansen, N., Treede, R.D., Klein, T.: The human pain system exhibits higher-order plasticity (metaplasticity). Neurobiology of Learning and Memory 154, 112--120 (2018)
[23] Robinson, B.L., Harper, N.S., McAlpine, D.: Meta-adaptation in the auditory midbrain under cortical influence. Nat. Commun. 7, 13442 (2016)
[24] Levy, D.A., Nail, P.R.: Contagion: A theoretical and empirical review and reconceptualization. Genetic, social, and general psychology monographs 119(2), 233--284 (1993)
[25] Treur, J.: On the Applicability of Network-Oriented Modeling Based on Temporal-Causal Networks: Why Network Models Do Not Just Model Networks. Journal of Information and Telecommunication 1(1), 23—40 (2017)
[26] Ashby, W.R.: Design for a Brain. Chapman and Hall, London (second extended edition). First edition, 1952 (1960)
[27] Port, R.F., van Gelder, T.: Mind as motion: Explorations in the dynamics of cognition. Cambridge, MA: MIT Press (1995)
[17] McPherson, M., Smith-Lovin, L., and Cook, J.M.: Birds of a feather: homophily in social networks. Annu. Rev. Sociol., 27, 415--444 (2001)
[18] Pearson, M., Steglich, C., Snijders, T.: Homophily and assimilation among sport-active adolescent substance users. Connections 27(1), 47—63 (2006)
[19] Sharpanskykh, A., and Treur, J.: Modelling and Analysis of Social Contagion in Dynamic Networks. Neurocomputing 146, 140--150 (2014)

[28] Carley, K.M.: Inhibiting Adaptation. In Proceedings of the 2002 Command and Control Research and Technology Symposium, pp. 1--10. Monterey, CA: Naval Postgraduate School (2002).
[29] Carley, K.M.: Destabilization of covert networks. Comput Math Organiz Theor 12, 51–66 (2006).
[30] Fessler, D.M.T., Clark, J.A., and Clint, E.K.: Evolutionary Psychology and Evolutionary Anthropology. In: The Handbook of Evolutionary Psychology, (D.M. Buss, ed.), pp. 1029--1046. Wiley and Sons (2015).
[31] Fessler, D.M.T., Eng, S.J., & Navarrete, C.D.: Elevated disgust sensitivity in the first trimester of pregnancy: Evidence supporting the compensatory prophylaxis hypothesis. Evolution & Human Behavior 26(4), 344-351 (2005).
[32] Hofstadter, D.R.: Gödel, Escher, Bach. New York: Basic Books (1979)
[33] Anten, J., Earle, J., Treur, J. : An Adaptive Computational Network Model for Strange Loops in Political Evolution in Society. Proc. of the 20th International Conference on Computational Science, ICCS'20, vol. 2, pp. 604–617. Lecture Notes in Computer Science, vol. 12138. Springer Nature (2020).
[34] Rojas, R.: Neural Networks. Berlin: Springer Verlag (1996).

Defect Detection of Knitted Fabrics Using Image Analysis

Ruvishka Rathnayake[1], , Chathura Rajapaksha[2],
[1,2] Department of Industrial Managment, Faculty of Science, University of Kelaniya, Sri Lanka.
[1] ruvishkarathnayake@gmail.com, [2] chathura@kln.ac.lk

***Abstract*—A fabric defect is defined as any abnormality that spoils the aesthetics of the fabric which hinders its acceptability by the consumer. The defects in the fabric can cause to reduce the finish garment's price by 45–65%. This research is motivated on detecting defects of weft-knitted fabrics which have a unique surface that may lead to holding back the human vision defects inspection. It triggers the emerge of automated fabric defect inspection to replace manual inspection while increasing accuracy and convenience. The model identifies the images by 'defected' or 'defect-free' using digital photos on the manufacturing premises. To avoid an overfitting dataset, data augmentation techniques are used and grey-scale images to minimize the complication with RGB images that cause to increase in the accuracy of the model with a larger number of data. The model performed 98.00% and 95.67% of training and validation accuracy rates respectively.**

Keywords— fabric defect detection, image classification, weft-knitted, Convolutional Neural Networks (CNN)

I. INTRODUCTION

The apparel manufacturing industry can be thought-out as the lifeblood of Sri Lankan's economy. The apparel and textile industry contributes 6% to Sri Lanka's GDP while accounting for 40% of the country's total exports by 2019 [1]. In the apparel business, fabrics are taken as the main material. Among different kinds of fabrics, knitted fabrics are known as comparatively faster and more economical process-wise which could open up new manufacturing opportunities for countries like Sri Lanka. Knitted fabrics are different from other fabric types by nature, appearance and are often superimposed since these materials are more flexible and can be more readily constructed into smaller pieces. The surface of a knitted fabric is also somewhat complex which can be easily identified. According to the knitting method, knitted fabrics are classified into two parties as weft-knitted and wrap-knitted.

"Quality" simply indicates that customer needs are satisfied at the end of the day. Ensuring the quality of a product is considered a crucial point in contemporary industrial manufacturing. When it comes to the apparel industry, the quality of the finished garment decides the future of the business without a doubt. The ultimate goal of the quality control process is to maximize the production of garments within the specified tolerances correctly the first time and to achieve a satisfactory design of the fabric or garment concerning the level of choice in design, styles, colors, the suitability of components and fitness of the product for the market. Any abnormality in the fabric that hinders its acceptability by the consumer is considered as fabric defect. The statistics indicate that fabric defects may cause to reduce the price of a finish garment's 45–65% [2].

The traditional method of human vision based defect detection could be helped in finding minor defects immediately. Nonetheless, the efficiency of manual detection can reduce gradually with the working time. Hence, it is necessary to develop an automatic inspection model for fabric defects to improve the quality of the fabric, while reducing human labor costs and errors that tend to occur. Though many types of research are out there that focus on the very problem, the apparel industry resists to acquire them due to the costly investment they have to make. Fundamentally, automated fabric inspection involves two challenges as defect detection and defect classification. The methods for fabric defect detection can be varied with various fabric types, manufacturing methodologies, etc. The defect classification also follows the same. Through this research, a model for defect defection of weft knitted fabrics will be proposed that can be implemented at a minimal cost and computational power using Convolutional Neural Networks (CNN) with high accuracy. The model is capable of identifying the defects and categorizing them basically into four categories

Fig. 1 Defect-Free Weft-knitted Fabric Images

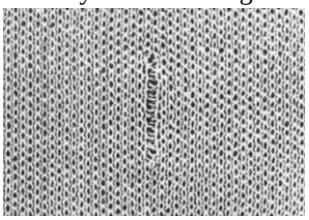


Fig. 2 Drop Stitch Defect of Weft-knitted Fabric

Fig. 3 . Holes Defect of Weft-knitted Fabric

Fig. 4 Oil Stains Defect of Weft-knitted

as defect-free, drop stitches, holes, and oil stains.

II. REVIEW OF PREVIOUS WORK

For defect detection and classification purposes, various techniques have been used in researches. They are driven through either online or offline basis. According to the chosen method for fabric defect detection, the researches can be regarded into seven approaches as structural, statistical, spectral, model-based, learning, hybrid, and comparison [3].

The structural approach mainly considers texture as a composition of textural primitives. The overall texture of the fabric pattern can be achieved with the composition of simple texture structures according to this method. In statistical approaches, first-order statistics and second-order statistics are used to extract textural features in

texture classification. Bi-level Thresholding, Gray level statistics, Morphological operations, Autocorrelation,

Normalized cross correlation, Multilevel thresholding rank order functions like techniques are used in this method. A large number of studies are conducted focusing on the spectral approach. Such researches need to have a high degree of periodicity. Some techniques such as Fourier transform Gabor filtering [14], Wavelet transform [15] is applicable to this approach. However, this approach is not recommended for fabrics containing random texture. In model-based approaches, with the help of the model built, identification of the texture as well as texture synthesis is carried out. Suitable for fabric images that have surface changes due to defects for instance yarn breakage and needle breakage. Under the learning-based approach, many pieces of research are there that cover up various techniques, naming a few, ANN, adaptive neuro-fuzzy system, Support Vector Machines (SVM), Gauss Markov random field, Poisons model, Model-based clustering, etc. By combining two or more approaches from above, some researches have used a hybrid approach. The main objective of using a hybrid approach is minimizing the computational complexity and increasing the rate of defect detection. Comparison studies are done by comparing the methods mentioned above which have great significance.

Some researches have been focused on fabric defect detection, have used Artificial Neural Networks (ANN), wavelet transform, mathematical morphology, Fourier transform, Gabor transform, etc as their technique. Zhoufeng Liu. et. al. has introduced a lightweight CNN to reduce the computational costs and storage services using depth-wise separable convolutions. For feature extraction, they have used a multi-scale feature extraction method. In the above research, the dataset consists of 3000 images and has achieved a 97.7% accuracy rate [4]. Research has been conducted based on uniform textured fabrics by Prasanna Bandara, et al. which used thresholding and morphological operations for feature extraction in fabrics under different light conditions for defect identification [5]. A novel detection algorithm based on Gabor-HOG (GHOG) for feature extraction and low-rank decomposition for decomposing the fabric image into normal background and defect, respectively has been introduced for Pattern fabrics (dot, box, and star-patterned). There's no guarantee in this method whether it will suit complex patterns as well since a limited number of patterns have been used [6].

The researches that have aimed the defect classification have been used some classification techniques like Bayesian classifier [13], Convolutional Neural Networks, Fuzzy C-mean method, and more. For example, the research base on the same domain, knitted fabrics, has done the classification of defects using Artificial Neural Networks which has mainly limited to holes and thick places in the knitted fabric. In this research, the best evaluation performance was obtained as 83.3% even with the limited defect types [7]. As the classification method, the Fuzzy C-Mean method is preferred in some researches. In this selected research, it has focused on the missing thread in weft or warp, oil stains, and holes kind of defects with a limited dataset of 45 samples [8]. Habib, et al. has conducted the research using a statistical approach for feature extraction and the Bayesian classifier to classify the revealed defects in the fabric. The results have reached 99.85% accuracy. The researchers considered 13 features from both geometric and statistic characteristics. As they mentioned, they completely relied on the dataset with features extracted with statistical techniques [9].

Related to the weft-knitted fabrics domain, there are a few kinds of research that have been conducted recently aiming at the knitted fabrics. The techniques varied from one another and even it can be varied with the knitting machines also. For instance, proposing Shearlet transform with segmented threshold de-noising to segment a warp-knitted fabric defect can be highlighted [10]. In another research, they have applied the same technique with single jersey knitting fabric defects using a comprehensive fabric database called Fabric Defect Detection Database (FDDD). This proposed system has achieved higher accuracy like 94% [11]. Bassel A. El-Azab, et al. targets to locate, classify defect of fabrics in order to control the circular weft knitting machine based on three critical weft-knitted fabric defects using log-gabor and neural network as a classifier with an acceptable accuracy of 80% [12].

III. METHODOLOGY

Methodology of this research is to implement a suitable model to identify defect detection and classify them accordingly with high accuracy and efficiency. The methodology carries out in five main phases as Image Acquisition, Image Preprocessing, Partitioning Data, Data Augmentation, and Training and Testing.

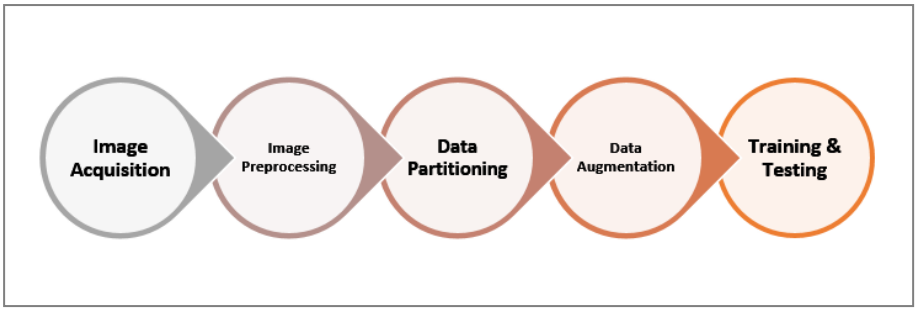


Fig. 1 Methodology of the Research

A. Image Acquisition

The inputs for the model would be weft-knitted fabric images that were taken from an 18MP digital camera from top-down view under white light beam (as per the previous literature this is proved as the best lighting condition for fabric defect detection [5]) at the manufacturing premises. The images are categorized as Defect-free, Oil Stains, Holes, and Drop Stitches where each category contains over 500 images in JPG format. The images are saved

under JPG format as the initial dataset for this research study. This image dataset includes 500 images from each as defective and non-defective weft knitted fabric images.

B. Image Preprocessing

The acquired data consists of different characteristics and are messy by nature. Those images should be standardized and cleaned up before feeding them to the neural network. Image preprocessing helps to reduce the complexity of the images that will speed up the training of the CNN. Images were converted into grayscale, which is a range of shades of grey without apparent color. This model range is represented from 0 - 1. RGB (red, green, blue) images contain more information than black and white images which will cause to add unnecessary complexity and take up more space in memory. This conversion of RGB images to grayscale images makes the dataset with 3 channels to 2 channels. Convolutional Neural Network accepts data as a matrix of numeric values. In order to do that image pixel intensities are converted into numeric values in the range of 0-1. Then those numeric values are divided by 255 to range the intensities to 0-1. It can be illustrated simply as below.

C. Data Partitioning

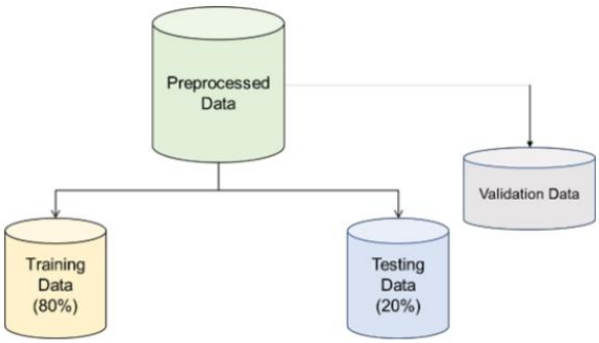


Fig. 2 Data Splitting Procedure

For the data partitioning, a classical statistical sampling technique called Simple Random Sampling (SRS) is used. In this research, the data will be divided into training, testing, and validation datasets. 10% of random images from the training dataset will be taken into the validation dataset where those will be used for validating the model trained. The rest of the data set is split into training (80%) and testing (20%) datasets.

D. Data Augmentation

Data augmentation is a technique that enables us to significantly increase the diversity of data available for training models by artificially creating from the existing data set. This is done by applying domain-specific techniques to images from the training data that create new and different training images. Image data augmentation is perhaps the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image.

Data augmentation can be achieved by transforming data cropping, padding, width-shift, height-shift, zoom, and horizontal flipping kind of techniques.

When training any neural networks, it is always encouraged to use a large volume of data. We account for these situations by training the neural network with additional synthetically modified data. Since the data set consists of 500 images from each defective and non-defective image, image data augmentation is used here to enhance the volume of data where a variety of algorithms for data augmentation can bring huge potential for improving data-greedy deep learning algorithms such as Convolutional Neural Networks. In this research data augmentation techniques like horizontal flip, vertical flip, scaling, image rotation by 450 and image zooming have been used to avoid data overfitting over the training and testing dataset.

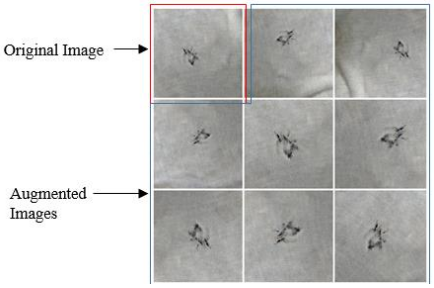


Fig. 3 Data Augmentation

E. Training and Testing

There are many different approaches for supervised learning, such as SVM, ANN, and CNN. The literature has shown that the CNN method performs best when it comes to feature extraction from images by itself and image Classification at a minimal cost. The labeled dataset will be used for training and testing the classifier model. The learning task is to predict the outcome of any valid input object after being trained to a satisfactory number of training examples. Therefore, Convolutional Neural Networks is chosen to perform the overall methodology of the study.

As mentioned earlier 500 images from four categories are used to train and test the model which were collected from weft-knitting manufacturing premises. After the training & testing using the CNN, model accuracy is calculated according to the following formula, with the validation dataset

IV. Experimental Results

Images of defect-free and other three types of defects were collected on knitting premises and a data set was created using those images.

As mentioned in previous literature, we were able to archive the mentioned accuracy level of 97.7% using the collected dataset with a multi-scale feature extraction method. The collected data of weft knitted fabrics was trained using a sequential CNN model using pre-processing techniques was able to archive a 98.00% of training accuracy and 95.67% of validation accuracy from the trained model. As in Fig. 3 and Fig. 4, it can be identifying that the model learns by reducing the training loss and increasing the training accuracy.

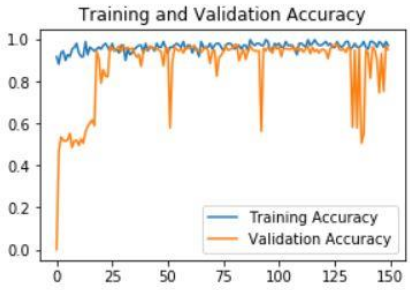


Fig.4 Training and Validation Loss for Trained CNN Model

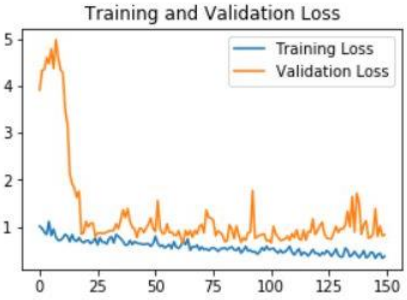


Fig. 5 Training and Validation Accuracy for Trained CNN Model

The following Table 1. shows the architecture of the developed sequential CNN for the defect detection and classification. The architecture was chosen considering the simplicity and the expected output follows the liner topology. The introduced sequential CNN consists of 12 layers which starts with an input layer. Rest of the layers consists as three set of layers of a convolutional layer [(16x2x2), (32x2x2), (64x2x2)] to extract the features from the input, a batch normalization layer to normalize the output of previous ReLU activation and accelerate the training and a max pooling layer to return the maximum value from the portion of the image covered by the Kernel. As the latter part of the model a flatten layer, dropout layer and a dense layer are used in this model.

$$\text{Accuracy (\%)} = \frac{\text{(Total number of correctly classified images)}}{\text{Total number of test images}} * 100 \quad \textbf{(1)}$$

Layer (Type)	**Param #**
conv2d_1 (Conv2D)	80
batch_normalization_1	64
max_pooling2d_1	0
conv2d_2 (Conv2D)	2080
batch_normalization_2	128
max_pooling2d_2	0
conv2d_3 (Conv2D)	8256
batch_normalization_3	256
max_pooling2d_3	0
dropout_1 (Dropout)	0
flatten_1 (Flatten)	0
dense_1 (Dense)	6725
Total params:	17,589
Trainable params:	17,365
Non-trainable params:	224

Table 1 Architecture of the Sequential CNN for Defect Detection and Classification

V. Conclusion and Future Work

In this study, the Sequential Convolutional Neural Network was used for creating the models, which were built according to defected and defect-free images. The trained model shows that this can identify the defected weft-knitted fabrics with 98.00% of accuracy using the images captured using an 18MP digital camera.

Moreover, the model can be implemented as a complete solution from image acquisition to image classification for the weft-knitted fabric manufacturing industry specifying with the knitting machine type as well. The research will be extended to a comparison of the developed model with built-in CNN architectures like ResNet50 and VGG16 for image classification to select the most suitable model that can give a higher accurate and efficient solution at a minimal cost

VI. References

[1] S.N. Niles , S. Fernando , W.D.G. Lanerolle, "A System for Analysis, Categorisation and Grading of Fabric Defects using Computer Vision," Research Journal of Textile and Apparel, vol. Vol. 19 No. 1, pp. 59-64, 2015.

[2] M. HAMZA, "Sri Lanka apparel exports to lose US$1.5bn as Coronavirus cripple buyers," economynext, 31 03 2020. [Online]. Available: https://economynext.com/sri-lanka-apparel-exports-to-lose-us1-5bn-as-coronavirus-cripple-buyers-63205/#.

[3] Hanbay, K., Talu, M. F., & Özgüven, Ö. F., "Fabric defect detection systems and methods—A systematic literature review," Optik, p. 127(24), 2016.

[4] Zhoufeng Liu, Jian Cui, Chunlei Li, Shumin Ding, Qingwei Xu, "Real-time Fabric Defect Detection based on Lightweight Convolutional Neural Network," 2019 8th International Conference on Computing and Pattern Recognition, 2019.

[5] Prasanna Bandara, Thilan Bandara, Tharaka Ranatunga, Vibodha Vimarshana, "Automated Fabric Defect Detection," 2018 International Conference on Advances in ICT for Emerging Regions, 2018.

[6] Chunlei Li, Guangshuai Gao, Zhoufeng Liu, Di Huang, Sheng Liu, Miao Yu, "Defect detection for patterned fabric images based on GHOG and low-rank decomposition," IEEE Access, pp. PP(99):1-1, 2017.

[7] Subrata Das, Amitabh Wahi, Sundaramurthy S, Thulasiram N, Keerthika S., "Classification of knitted fabric defect detection using Artificial Neural Networks," 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2020.

[8] JMALI Mohamed, ZITOUNI Baghdadi and SAKLI Faouzi, "Classification of fabrics defects using image analysis and fuzzy c-means method," International Conference Of Applied Research On Textile, Citrat-6, 2014.

[9] Habib, M. T., Shuvo, S. B., Uddin, M. S., & Ahmed F., "Automated textile defect classification by Bayesian classifier based on statistical features," 2016 International Workshop on Computational Intelligence, 2016.

[10] Zhijia Dong, Dong Xia, Pibo Ma, Gaoming Jiang, "Warp-knitted Fabric Defect Segmentation Based on the Shearlet Transform," Fibres and Textiles in Eastern Europe, pp. 25(5):87-94, 2017.

[11] Kazım Hanbay, Muhammed Fatih Talu, Ömer Faruk Özgüven, Dursun Öztürk, "Real-Time Detection of Knitting Fabric Defects Using Shearlet Transform," Tekstil ve Konfeksiyon , p. 29(1), 2018.

[12] Bassel A. El-Azab, Marwa Yasseen, Magdi Mohamed Fahmi, Heba Shalaby, R.A.M. Abd El-Hady, "Detection And Classification of Weft Knitted Fabrics Defects Using Gabor Wavelet," International Journal of Advances in Scientific Research and Engineering (ijasre), 2020.

[13] Daniel Yapi, Marouene Mejri, Mohand Sa¨ıd Allili, Nadia Baaziz, "A Learning-Based Approach for Automatic Defect Detection in Textile Images," IFAC (International Federation of Automatic Control), 2015.

[14] K. L. Mak, P. Peng, H.YK. Lau, "A Real-time Computer Vision System for Detecting Defects in Textile Fabrics," IEEE International Conference on Industrial Technology, 2005.

[15] H.İ. Çelika, , L.C. Dülger, M. Topalbekiroğlu, "Development of a machine vision system: real-time fabric defect detection and classification with neural networks," The Journal of The Textile Institute, vol. Vol. 105, no. No. 6, p. 575–585, 2014.

Tilly – A Tamil Learning Chatbot for Non-Native Tamil Speakers

M.A.S.T Goonatilleke[1], M.W.G Jayampath[2], B Hettige[3]
[1,2,3] Intelligent Research Laboratory, Faculty of Computing,
General Sir John Kotelawala Defence University, Sri Lanka

***Abstract*— Sri Lanka is a multi-ethnic, multilingual country, and Sinhalese, Tamils are considered the major ethnic groups in the country. These two ethnic communities maintain their mother languages, Sinhala and Tamil that are official languages in communication and for official purposes. As Sri Lankans, learning both official languages are essential and important to maintain harmony and peace among people. Although Sinhalese and Tamils have lived together for a long time, it is an unfortunate situation that they are unable to communicate with both official languages. This problem has severely affected the majority Sinhala community in the country due to their low level of Tamil language knowledge. To overcome this current situation, most people are trying to learn the Tamil language, and learning such as a new language has become a difficult and challenging task. As a solution, the Chatbot system can be introduced which is a very popular interactive communication media between humans and machines. This paper presents the design and implementation of a Tamil chatbot system named Tilly to learn the Tamil language. Using this system people can ask questions in Tamil language and the system responds to the user in the same language. The question base of the system has been implemented based on three categories of questions such as simple, moderate and advanced depending on the ability of each person in the Tamil language. Accordingly, any person with any level of Tamil language knowledge can use this system to improve the language fluency as a hobby in their day-to-day lives. This system has been designed using Dialogflow and Google cloud platforms. Finally, the system has been tested in a practical environment using human evaluators and successful results were obtained**

Keywords*— *Chatbot, Tamil, Dialogflow, Google cloud platform

I. INTRODUCTION

Sinhalese, Tamils are the major ethnic groups in Sri Lanka and they have a long history that is going back more than 2000 years [1]. These two major linguistic communities maintain their mother languages such as Sinhala and Tamil in communication and also use them for official purposes in their territories. [2] In the face of this situation, the Government of Sri Lanka has paid special attention to these two languages and has adopted legislative policies for them. Accordingly, the 13th Amendment to the Constitution of Sri Lanka has made Sinhala and Tamil as official languages and in 1988, the 16th Amendment to the constitution corrected the position by stating, "Sinhala and Tamil shall be the languages of administration throughout Sri Lanka." [3]

The pathetic problem of these two official languages in the country has arisen with the linguistic survey that was conducted in 2010. The survey mentioned that around 90% of Sinhala speaking people are unable to communicate in the Tamil language and 70% of Tamil speaking people are unable to communicate in the Sinhala Language [4]. Accordingly, this is a very serious and a pathetic problem because, although these Sinhalese and Tamils are living together for many centuries, learning both official languages were not achieved. Besides, learning Sinhala and Tamil languages is a very essential fact to maintain peace, harmony among two communities [24]. The report produced by the Lessons Learned and Reconciliation Commission (LLRC) in 2011, appointed by the Sri Lankan Government includes some recommendations regarding these languages as follows [5].

The learning of each other's languages should be made a compulsory part of the school curriculum. This would be a primary tool to ensure attitudinal changes amongst the two communities. Teaching Tamil to Sinhala children and Sinhala to Tamil children will result in a greater understanding of each other's cultures.

The proper implementation of the language policy and ensuring trilingual (Sinhala, Tamil, and English) fluency of future generations becomes vitally important. A trilingual education will allow children from very young days to get to understand each other.

Due to this situation, the Government of Sri Lanka recommends learning all three languages such as Sinhala, Tamil, English and make the learning of official languages is compulsory for all the public servants. Moreover, learning all three languages is included in the school curriculum also. However, this problem has severely affected the majority Sinhala community in Sri Lanka due to their low level of Tamil language knowledge. To overcome this problem, most of them are trying to learn the Tamil language and it has been in operation for some time now.

Generally, studying and learning such a new language is a difficult, challenging task and requires timely and practical methods. But, in a rapidly developing technological advanced world, these tasks can be solved very efficiently with the help of novel and emerging technologies. The Chatbot concept can be introduced to overcome this challenge practically and enjoyably which is one of the most emerging technologies in the 20th century [17].

This paper is going to discuss a Tamil learning Chatbot named Tilly that was designed for Non-Native Tamil Speakers in Sri Lanka. This system operates in Tamil language and the user can maintain the communication in the Tamil language. This chat system contains more than 250 question types and is categorized into three domains as simple, general questions, moderate questions and advanced questions. Accordingly, a person having different levels of Tamil language abilities can use this system as an entertainment tool to improve the language fluency. A user with a week knowledge of Tamil language ability, can start the communication between the Tilly with simple questions and gradually move to the other two domains. This system allows any user to learn the Tamil language in a practical, enjoyable and effective way. Simply, this system acts as an assistant to learn the Tamil language in an enjoyable way. Accordingly, any person who is willing to learn Tamil can use this system to improve the language fluency as a hobby in their day-to-day lives.

The rest of this paper is organized as follows. Section 2 explains Chatbot systems and its importance. Then Section 3 gives details about existing Chatbots and Section 4 describes the Dialogflow technology that was used to develop the Tilly system. After that, Section 5 reports the design and the implementation of the developed Chatbot system and finally, Section 6 concludes the paper with a note on further works.

II. Chatbot – An Emerging Technology of the 20th Century

Chatbots are considered to be an emerging technology in the world of the 20th century. It is an artificial intelligence software that is designed to conduct a conversation or a chat using several techniques including auditory and textual, especially over the internet [18]. To achieve this, the Chatbots provides an interface between the computer and the user with some intelligent features. A Chatbot can be also known as a smart bot, chatterbot, talkbot, IM bot, and interactive agent which is one of the most advanced and promising expressions of interaction between machines and humans [6]. When considering the history of Chatbots, it has been started in 1950 as a result of the Turing Test, and ideas of that test essentially laid the foundation for the revolution of Chatbots. Today Chatbots are often used in the field of business and many reasons were identified. Accordingly, many facts motivate people to interact with them as it increases productivity by providing facilities to access information quickly and efficiently. It gives entertainment when users have nothing to do as well as to avoid loneliness. Besides, it increases the curiosity of people.

There are many classifications in Chatbots. Among them, this paper focuses on two major classifications. The first classification depends on how the specific bots were programmed. Accordingly, they can be divided into two groups as simple Chatbots and smart or advanced Chatbots [8]. Simple Chatbots work according to pre-prepared commands which have been written by the developer using regular expressions while smart or advanced Chatbots are based on artificial intelligence when they communicate with users. The smart Chatbots are not based on pre-prepared answers like simple Chatbots, and they respond with adequate suggestions. The second classification is standalone chat systems and web-based chat systems. Among computer-based or standalone chat systems is the most popular method used today. Not only that but also

Figure 1 Growth Rate of Chatbots

these chat systems can be again divided into two categories named human-computer dialogue systems and human-human dialogue systems [25]. Today, there are so many human-human chat systems available in the world such as MSN Messenger and Yahoo Messenger. They are the most popular chat systems in the world. Also, developing a human-computer chat system including natural language capabilities is a challenging task and it is as old as the field of Artificial Intelligence.

It is very interesting to discuss the importance and advantages of Chatbots. A Chatbot is a software program, powered by rules and Artificial intelligence. It provides a chat interface that simulates a real interaction with users [9]. Today Chatbots have become a very popular communication and business tool. Accordingly, the utilization of Chatbots has certainly gained momentum in the last five years. Figure 2 shows the growth rate of Chatbots in the last few years.

A. Customer Service.

When compared with other areas, Chatbots have become very popular in the business sector. As a result, most of the websites provide a small live chat tool that helps customers to find information. Besides, these customer service Chatbots are available in mobile phones for handling customers and help to save money of the customers and also to provide automation for repetitive work.

B. Education

The Chatbots can make education more accessible and more engaging as it is a very attractive tool. The students who are using educational Chatbots can send questions in-order to get answers for a particular subject area. Accordingly, it can also be used as a quiz game.

C. Mental Health

Chatbots can be used as entertainment tools to maintain balanced mental health. Generally, music Chatbots can help to create a more personalized playlist based on user preferences. Also, using Chatbots, online gambling operators can utilize smart assistants to manage live chats, analyze odds, or even distribute custom bonuses following the player status.

III. Related Works

In 1950, there was a generation with scientists, mathematicians, and philosophers with some concepts

including intelligent machines and artificial intelligence. Alan Turing, who was a young British polymath, explored the mathematical possibility of artificial intelligence. Not only that but also one of his papers named Computing Machinery and Intelligence has discussed how to build intelligent machines as well as how to test their intelligence. This is considered as the first idea of intelligent machines. Besides, Chatbot is also an intelligent system that includes intelligent interfaces [23].

The history of Chatbots was started in 1950 as a result of the Turing Test, and ideas of that test essentially laid the foundation for the revolution of Chatbots. Then, many Chatbots have come to the stage such as Eliza (1966), Parry (1972), Jabberwacky (1988), Dr Sbaitso (1992), Alice (1995), Smarterchild (2001), IBM's Watson (2006), Siri (2010), Google Now (2012), Alexa (2015), Cortana (2015), Bots for Messenger (2016) and Tay (2016) [7]. A brief description of some of the common chatbots is given below.

A. Eliza

ELIZA has been created by Joseph Weizenbaum, during the period from 1964 to 1966. It is an early natural language processing computer program that has been created at the MIT Artificial Intelligence Laboratory. It was one of the first Chatbots as well as the first program capable of attempting the Turing Test. This Chatbot was created to simulate the communication between humans and machines by using a pattern matching [10].

B. Alice

Alice or simply Alicebot is a natural language processing Chatbot that holds a conversation with a human by applying a heuristical pattern matching technique to identify user input. Not only that but also, this algorithm used depth-first search techniques. ALICE stands for Artificial Linguistic Internet Computer Entity that was composed of Richard Wallace in 1995 [11]. In 1998 the program was written in Java and used an XML Schema called AIML (Artificial Intelligence Markup Language). Also, ALICE has passed the Turing test in two consecutive years.

C. Mitsuku

Mitsuku was created in 2012 by Steve Worswick. It was created using AIML technology and has won Loebner Prize in 2013, 2016, 2017, and 2018. Mitsuku contains most of Alice's AIML files [12].

D. Cleverbot

Cleverbot is also a Chatbot system that uses the Artificial Intelligence algorithm. It a web application and was created by Rollo Carpenter, a British AI scientist. Not only that but also Cleverbot is also available in several different operating systems such as iOS, Windows, and Android. Unlike some other chatterbots, Cleverbot's responses are not pre-programmed [13]. From 2014 Cleverbot has been upgraded to use GPU serving techniques and developers are attempting to build new versions using techniques of Machine Learning.

E. Octopus: A Multi-Agent Chatbot

Octopus is a multi-agent-based Sinhala chatbot system that was implemented through the MaSMT framework. This chat system contains 8 sub-multi-agent systems namely core system, GUI system, Natural Language Processing system, communication system, learning system, action system, searching system and data access system to handle intelligent capabilities of the chatbot. Octopus has been implemented through Java and can run on two operating systems such as Windows and Linux [20].

F. Sinhala Chatbot

Hettige and Karunananda have developed a Sinhala Chatbot in 2006. This was developed using Prolog and based on Natural language processing modules. These modules are Sinhala parser and Sinhala Morphological analyzer. Besides the entire Chatbot system has been developed using SWI-PROLOG and JAVA that can run on both Windows and Linux operating systems [14].

G. POONGKUZHALI-An Intelligent Tamil Chatterbot

Kalaiyarasi and Parthasarathi have developed an intelligent Tamil Chatbot named POONGKUZHALI. The developed chatbot is a system that simulates human conversation through Artificial Intelligence using the Tamil language. A question or a statement from the user is taken as the input and the system generates an appropriate response as the output based on the context of the input. The response is formed using a set of reassembly rules that reside in the knowledge base. After that, the response is reframed to match how the user had framed his question. Also, using this system, the user can choose any existing topic for conversation. This chatbot system has been designed using a set of decomposition rules [19].

Human-Computer text conversation through NLP in Tamil using Intent Recognition

Shanmugam U and others have developed a chatbot that leverages NLP by using the Dialogflow API. This chatbot has been mainly aimed to bridge the gap between the parents and educational institutions. The developed chatbots can respond to the parent's queries in the Tamil native language thus assisting parents in keeping track of their students' performance at college. The significant components of this Chabot incorporate intent detection, routines for conversation management, dialogue design and slot fulfilment. Besides, it also uses Flask micro framework and Pusher Channels [21].

IV. DIALOGFLOW

Dialogflow is a Google-owned framework that enables users to develop human-computer interaction technologies that can support Natural Language Processing (NLP) [15]. On the other hand, it is an end-to-end, build-once, and deploy-everywhere development platform that creates conversational interfaces for websites, mobile applications, messaging platforms, and IoT devices [16]. The architecture of the Dialogflow has 6 main components as Dialogflow console, agents, intents, entities, contexts, and events. These concepts are very essential when developing a chatbot or a voice bot.

A. Dialogflow Console

Dialogflow console is a web user interface that is used to create, build, and test agents. However, this Dialogflow Console is different from the Google Cloud Platform (GCP) Console as the Dialogflow Console is used to manage Dialogflow agents, while the GCP Console is used to manage GCP-specific Dialogflow settings. Generally, designers use the Dialogflow console to build agents but in advanced scenarios, Dialogflow API also can be used. Besides, using the console can perform several tasks.

Create agents, intents, and entities.

Control conversation paths with contexts.

Adding events that are triggered by occurrences outside the conversation.

Implement fulfilment to connect your service when using integrations.

Integrate with other conversational platforms.

Analyze agent performance.

The test developed the agent via the simulator.

B. Agents

A Dialogflow agent can be described as a virtual agent that handles conversations with users. The agent is very similar to a human call centre agent. It is a natural language understanding module that is capable of understanding the nuances of human language. When the user sends a text or audio during a conversation, Dialogflow translates that to structured data. Accordingly, the main purpose of agents of the Dialogflow is to handle many types of conversations required for the system. In Dialogflow, there are several types of agents such as prebuilt agents that are provided by Dialogflow for common use cases, multilingual agents that support many languages, and mega agents.

C. Intent

An intent categorizes the user's intention for one conversation turn. For one agent, designers can define many intents, and using combined intents can handle a complete conversation. When a user writes or something, it is referred to as "end-user expression". This expression will match the best intent of the voice bot by Dialogflow. This intent matching is known as intent classification. The basic intent contains four main parts as training phrases, action, parameters, and responses.

Training phrases – Training phrases are the example phrases for what a user might say. When a users' input expression resembles one of these training phrases, Dialogflow matches the intent. The advantage of this function is designer doesn't want to define every possible example.

Action: An action can be defined for each intent in the system. However, when an intent is matched, Dialogflow provides an action to the system. Besides, actions can be used to trigger certain actions that are defined in the system.

Parameters: Dialogflow provides extracted values from the user expression as parameters when an intent is matched at runtime. Each parameter has a type, named as entity type that dictates exactly how the data is extracted. Unlike raw user input, these parameters are structured data that can easily be used to perform some logic or to generate responses.

Responses: Responses are the output that should be

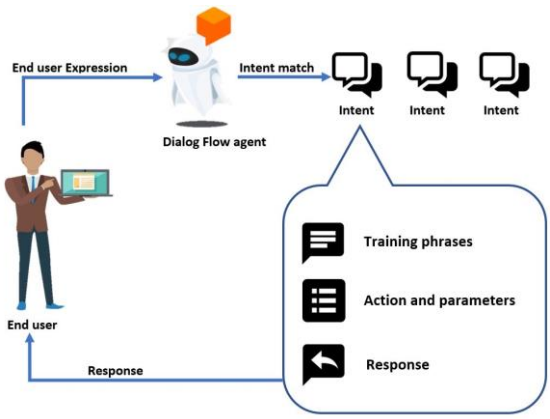


Figure 2 Basic Flow of the Intent Matching and Responding to the User

sent to the user as text, speech, or visual form. For every input that is sent by the user, the system has to send a response. These may provide the user with answers, ask the user for more information, or terminate the conversation.

D. Entities

Every intent parameter has a type named entity type. The entity type dictates exactly how data from user input or expression is extracted. Besides, Dialogflow already provides predefined entities that are named as system entities that match many common types of data. Therefore, there are system entities for matching colours, email addresses, dates, times, etc. Not only that but also, designers can create his/her custom entities to match custom data.

E. Contexts

Dialogflow contexts are similar to natural language contexts. For example, if a person says "they are apples", the system has to context to understand what "they" are referring to. Accordingly, to handle a user expression, it is essential to provide context to correctly match an intent. Also, using contexts, can control the flow of the conversation and can configure contexts for intent by setting input contexts and output contexts. These input and output contexts are identified by string names. When the contexts are active, Dialogflow matches intents that are configured with input contexts. Using contexts, you can control the flow of a conversation. You can configure contexts for intent by setting input and output contexts, which are identified by string names. When an intent is matched, any configured output contexts for that intent become active. While any contexts are active, Dialogflow is more likely to match intents that are configured with input contexts that correspond to the currently active contexts.

Events

Intents are matched when a user's input expression matches with an intent training phrase. Besides, can trigger intents using events and these events can be invoked in

several ways. There are two types of events such as platform events and custom events.

Platform Events: Platform events are built-in events that are provided by platform integrations. Generally, these events are invoked when platform-specific events occur.

Custom Events: Custom events are the events that are defined by the designer and can invoke them using fulfilment or the API.

V. Design and Implementation of Tilly

Tilly chatbot has been designed using Dialogflow and Google cloud platforms. It is one of the natural language understanding platforms that makes it easy to design and also to integrate conversational user interfaces. Dialogflow helps to build Chatbots and conversational IVR that enable natural and rich interactions between users. Using Dialogflow as the development platform, designers and users can earn a lot of advantages such as 24x7 availability, can design customized chatbots for different areas, gives immediate responses for inputs, supports 20+ global languages, can be integrated easily with may devices, etc.

The developed Tamil Chatbot system has been established to fulfil a question-answering purpose. Mainly, there are two persons in this system. The first person is the user who is a human being and the second person is the agent in the Chatbot system who acts like a human. Simply, the user asks questions in the Tamil language while the agent answer to them using the same language and simulates a chat between two persons. This system works according to the client-server process and the agent of the system works as the server. Figure 2 shows the basic architecture of the developed Tamil Chatbot system.

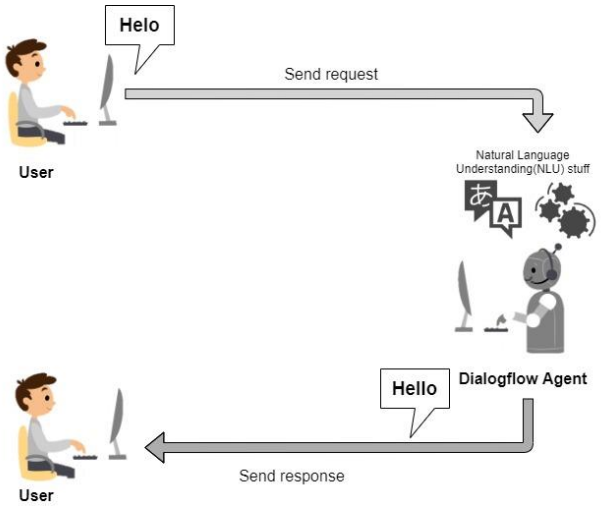


Figure 3 The Basic Architecture of the Tamil Chatbot

At first, the user of the system who is a human sends a text input message to the system using the Tamil language. After that this input will be taken by the system and apply NLU concepts and understands the received message. After that, it starts to generate a proper reply to the input and system sends that output text in the Tamil language to the user. As the final step, the user receives an output message from the system.

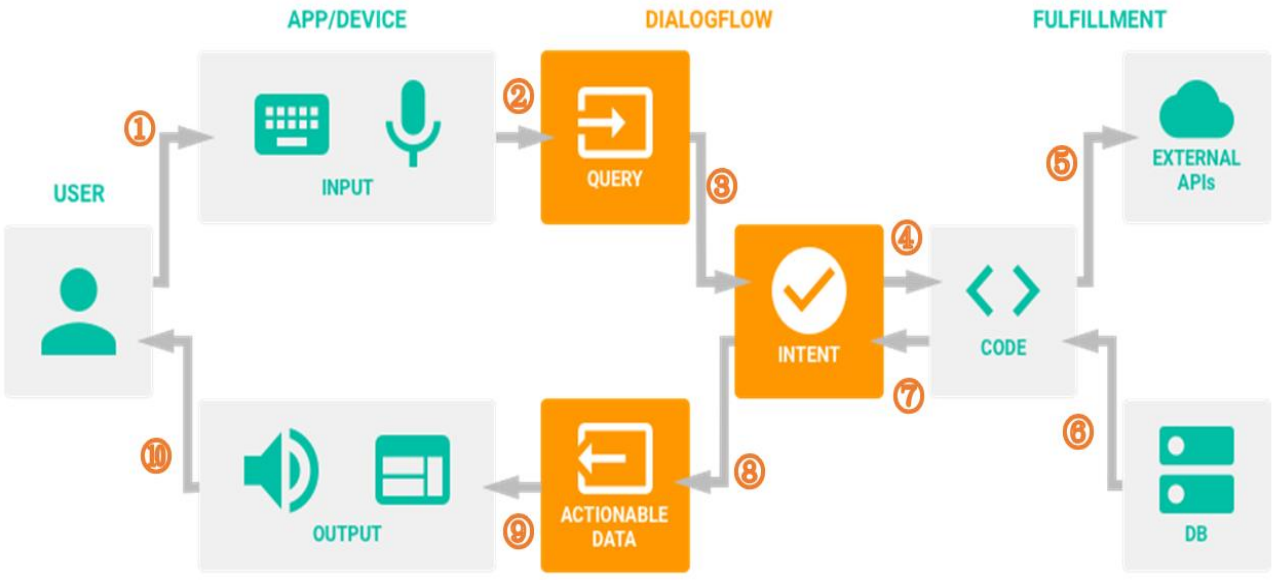


Figure 4 Design of the Tilly Chatbot

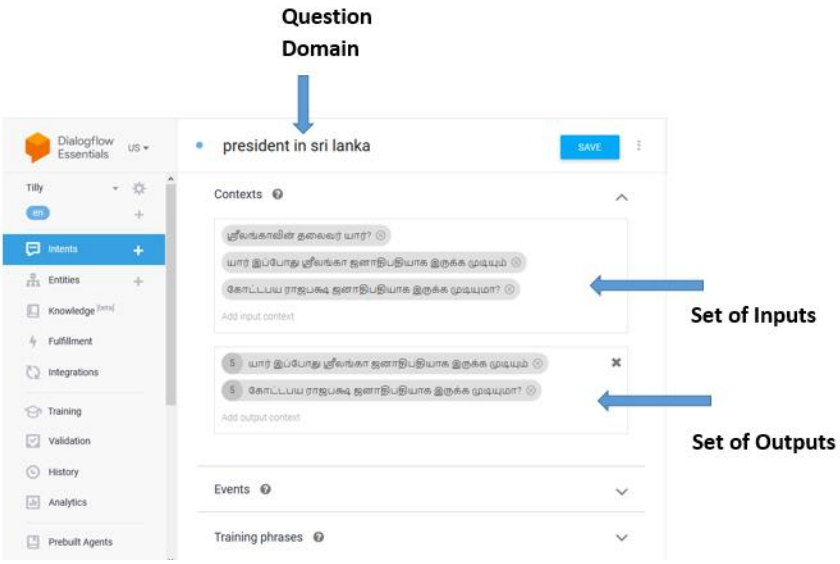


Figure 5 Declaration of Inputs and Outputs for Question Type

The system has been designed for any person having different levels of Tamil language ability. Based on the language ability, the question base of the system has been implemented for three types of questions such as simple questions, moderate questions, and advanced questions. Category 1 or simple question category has been designed with 100 question types for the people who have very low language ability. The questions of this category were limited to a maximum of two words. Secondly, category 2 has been designed with 100 question types and was designed for the people who have a moderate level of knowledge. The questions of this category were limited to a maximum of four words. At last, category three was developed with 50 advanced question types that are very useful for people who have very good language fluency. Accordingly, this chatbot supports 250 different question types and as a result, the system contains and has been trained for 250 different intents. Figure 3 shows the design of the system.

At first, the user sends a text message to the Chatbot agent named Tilly. This text message is the input of the system. In the second step, the text message will be taken by the Dialogflow agent named Tilly as a query. Then, the agent starts to categorize and match the received query with a corresponding intent that is already manually defined by the developer. The intent section has 250 different intents based on the question types. If the query is matched with an existing intent of the system, then the agent generates an actionable data which is an appropriate output and sends that to the user. However, when the query is not matched with one of the 250 existing intents of the system, the agent has to communicate with the fulfilment. This case will happen when the replies to the input are dynamic and the server of the system needs to retrieve data. Accordingly, Dialogflow provides webhook integration free of charge to allow Chatbot communication with the server. That means a response can be retrieved from the database using a webhook to provide a better reply to the user to his input without dealing with existing intents in the Dialogflow system. Accordingly, Webhook sends a formatted response to the intent. After that intents generate actionable data according to different channels and send that response to the user as the output.

The intent matching is the core activity in this system. That means to generate the correct and appropriate answer or the output for the user's input. This activity is mainly based on Natural Language Processing principles. The system has been trained with sample question patterns and answers. For example, a question type can be declared for the asking name of the president of Sri Lanka. Accordingly, the user can ask the same question in different ways with the same meaning. However the user asks a question in different ways, the system matches the received input with existing intents and generate the correct answer. Figure 4 shows the sample intent matching scenario for a particular question type. The question type which was selected is asking the name of the president of Sri Lanka. At first, have to add a set of sample user input questions that the user might be asked. These inputs should be added with the same meaning but in different forms. Then, should add a set of appropriate answers for the input questions.

The Tamil Chatbot system has been designed using Dialogflow and Google cloud platforms. Dialogflow is originally named as api.ai which is a Google product used to develop Chatbots using the Natural Language Understanding (NLU) concepts. This means Dialogflow can convert the user input or query to intent by using NLP and NLU concepts. Therefore, it helps to integrate with many IM (Intelligent Machines) platforms such as Line, Twitter and Facebook.

Figure 7 Evaluation Results of Test

The implementation process of this Chatbot takes several important steps. Among them, the first step is building a Firebase project at the Dialogflow console. Then, import the preconstructed agent named "Translate". When the designer imported the “Translate”, some intents are automatically created and do not want to create any intents and entities. Then as the next step, initialize a new project with functions using the Firebase Command Line Interface. Then, select the Firebase project which was created earlier. After that, install the Actions on Google and Cloud Translate API libraries for the node using the below code.

```
cd functions
npm install actions-on-google @google-cloud/translate
```

Then, edit “functions/src/index.ts” and import the code in the project of the Dialogflow from index.ts with removing its contents. As the next step, deploy the “deploy-output” function with firebase and copy the below URL function.

`https://us-central1-your-project.cloudfunctions.net/fulfill`

Then, select "Fulfillment" trigger the webhook and insert the feature URL into the URL field in the Dialogflow console. After that choose "Properties" and select it. After selecting “Properties” check the "Use webhook" under "Fulfillment" for any purpose that was imported and also save it. Then, enable the Cloud Translate API in the Google Cloud console for the Firebase project. If you don't enable this API, then the first time the translate API is called by your function and it will leave an error in the Functions log with a URL to follow to enable the Google Cloud Translate API. Copy that URL and load it into your browser to enable the API.

Figure 5 shows the User Interface of the developed Tamil Chatbot system. At the first user can starts with simple question collection and then moves to moderate and advance question types gradually.

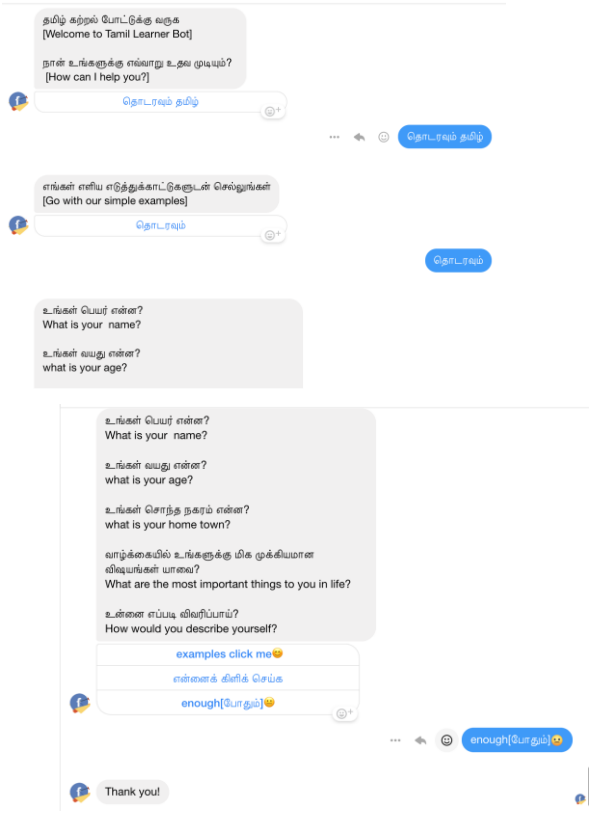


Figure 6 User Interface of the Developed Tamil Chatbot

Table 1 shows the results obtained from the test

Parameter	Very Good (5)	Good (4)	Moderate (3)	Poor (2)	Very Poor (1)
The clarity of the User Interface	21	9	0	0	0
Ease of Use	15	10	5	0	0
Validity of responses	9	15	3	2	1
Clarity of responses	9	9	6	3	3
Usefulness	15	14	1	0	0
Efficiency	8	14	4	3	1
Reliability	10	14	3	2	1

VI. Testing and Evaluation

The Tilly chatbot has been tested using human evaluators in a practical environment. At first, people have been selected with different levels of Tamil language literacy and they were divided into three groups of 10 each such as poor Tamil literacy level, moderate level and advanced level respectively, As the next step, the system was freely distributed among them and asked them to chat with Tilly. After that, they were asked to score for some parameters.

VII. Conclusion & Further Works

Sinhalese and Tamils have a long history in Sri Lanka that is going back more than 2000 years. These two major linguistic communities maintain their mother languages such as Sinhala and Tamil in communication and also use them for official purposes in their territories. However, unfortunately, both Sinhalese and Tamils can't communicate with both languages while they are living together for many years. Besides, the government of Sri Lanka also has ordered to learn both Sinhala and Tamil languages. Accordingly, a practicable solution can be introduced combined with new technologies to overcome this problem. This paper presented the design and the implementation of a Tamil learning Chatbot system named Tilly that can be used as an entertainment tool for non-native Tamil. This system has been successfully developed using Dialogflow, Google cloud platforms and tested using human evaluators in a practical environment and encouraging results were obtained. As further work can improve this developed system as a trilingual learning chatbot to learn the Tamil language.

References

[1] Statistics.gov.lk, http://www.statistics.gov.lk/page.asp?page=Population%20and%20Housing

[2] Ep.gov.lk, http://ep.gov.lk/en/13th-amendment-to-the-constitution

[3] Lawnet.gov.lk, https://www.lawnet.gov.lk/1947/12/31/sixteenth-amendment-to-the-constitution-2/

[4] Sabaratnam. A Teaching of Second National Language (Tamil) in Sri Lanka, http://ijebmr.com/uploads2018/IJEBMR_02_215.pdf

[5] The Asia Foundation, http://asiafoundation.org/2013/01/16/in-post-conflict-sri-lanka-language-is-essential-for-reconciliation/

[6] Forbes.com, https://www.forbes.com/sites/blakemorgan/2017/03/09/what-is-a-Chatbot-and-why-is-it-important-for-customer-experience/

[7] Medium, https://Chatbotsmagazine.com/a-visual-history-of-Chatbots-8bf3b31dbfb2

[8] Medium, https://medium.com/swlh/what-is-a-Chatbot-and-how-to-use-it-for-your-business-976ec2e0a99f

[9] Digital Doughnut, https://www.digitaldoughnut.com/articles/2017/october/top-7-benefits-of-Chatbots-for-your-business

[10] En.wikipedia.org, https://en.wikipedia.org/wiki/ELIZA

[11] En.wikipedia.org, https://en.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity

[12] Pandorabots.com, https://pandorabots.com/mitsuku/

[13] Cleverbot, https://www.cleverbot.com/

[14] Hettige B, Karunananda A, First Sinhala chatbot in action, Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI), University of Moratuwa (2006)

[15] Cloud.google.com, https://cloud.google.com/dialogflow/docs

[16] Wikipedia.org, https://en.wikipedia.org/wiki/Dialogflow

[17] Toptal Insights Blog, https://www.toptal.com/insights/innovation/chatbot-technology-past-present-future.

[18] Expert System, https://expertsystem.com/chatbot/

[19] 19. Kalaiyarasi.T and Parthasarathi. R, POONGUZHALI-An Intelligent Tamil Chatterbot, https://www.semanticscholar.org/paper/POONGKUZHALI-An-Intelligent-Tamil-Chatterbot-Kalaiyarasi-Parthasarathi/85010b17f256edf8e482aa5765adf4e8baa01d76?p2df.

[20] Hettige B, Karunananda A, Octopus: A multi-agent chat-bot, Proceedings of 8th International Research Conference KDU (2015).

[21] Shanmugam. U Mani.S, Sivakumar.S, Rajeswari. P, Human-Computer text conversation through NLP in Tamil using Intent Recognition, 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1-5, doi: 10.1109/ViTECoN.2019.8899571, Vellore, India (2019)

[22] Drift.com, https://www.drift.com/learn/chatbot/why-are-chatbots-important/

[23] Zemčík, Tomáš, A Brief History of Chatbots. DEStech Transactions on Computer Science and Engineering. 10.12783/dice/aicae2019/31439 9 (2019).

[24] Refworld.org, https://www.refworld.org/docid/501005892.html.

[25] Hussain, Shafqat, Omid, Ababneh, Nedal, A Survey on Conversational Agents/Chatbots Classification and Design Techniques (2019).

Spatiotemporal Ground Reaction Force Analysis using Convolutional Neural Networks to Analyze Parkinsonian Gait

Musthaq Ahamed [1] , P.D.S.H. Gunawardane[2], Nimali T. Medagedara[3]
[1,3] The Open University of Sri Lanka, Nawala, Nugegoda.
[2] The University of British Columnbia, Vancouver, Canada.
Email: [1] musthaqahamed02@gmail.com, [2] hiroshan@mail.ubc.ca, [3] tmmed@ou.ac.lk

***Abstract*— Parkinson's disease (PD) is a non-curable disease that commonly found among elders that greatly reduce their quality of life. PD primarily affects the gait pattern and slowly changes the walking gait from the normality to disability. The early diagnosing of PD is important for treatments and gait pattern analysis is used as a technique to diagnose PD. The present paper has identified the raw spatiotemporal ground reaction force (GRF) as a key parameter to identify the changes in human gait patterns associated with PD. The changes in GRF are identified using a convolutional neural network through pre-processing, conversion, recognition, and performance evaluation. The proposed algorithm is capable of identifying the severity of the PD and distinguishing the parkinsonian gait from the healthy gait. The technique has shown a 97% of accuracy in automatic decision-making process.**

Keywords*— Convolution neural networks, Deep learning, Parkinsonian gait analysis, Parkinson's disease

I. INTRODUCTION

Parkinson's disease (PD) is a progressive neurodegenerative disorder that affects elderly population. The PD extirpate a small region in the brain that controls movement, balance, and posture. In 2016, 6.1 million people are diagnosed with PD [1]. The reports have shown a significant increase in reported Parkinson cases due to the aging of the world population. In first five to ten years, most of the Parkinson subjects have not shown any significant symptoms to early diagnose this disease [2].

The available techniques to diagnose PD are relying on the experience of human technicians [3]. Unified Parkinson Rating Scale (UPDRS) is the most common technique that is used to diagnose PD [4]. The UPDRS consists of five categories to diagnose the disease severity stages including behavior, physical movement (motor) examination, mood, mentation, and day to day activities. In one of the studies, PD is evaluated by interviewing the subjects [5]. However, the analysis of gait characteristics/patterns for PD is found by Knutsson in 1972 [6]. This study has shown that the parkinsonian gait has a large variability compared with healthy gait.

Moreover, PD subjects have lower walking speed with a large gait cycle. The parkinsonian gait's stride length is shorter compare to the healthy gait. This specificity in the gait pattern is caused by the freezing of the gait and it is globally identified as a vital feature of identifying the PD [7]. Even though human gait movement and their posture are unique, in some cases these technicians have made mistakes in distinguishing parkinsonian gait [8]. Therefore, gait analysis of subjects with PD requires a highly sophisticated approach to get reliable results. Deep learning approaches are used to enhance the reliability in gait analysis [9].

In recent years, artificial neural networks and deep learning has been advanced to process and analyze complex data sets [10], [11]. The development of computer technologies has introduced new approaches for data processing, analyzing, and data classifying in healthcare applications [12]. The deep learning is used to automate the development of models that is used to analyze complex ground reaction force (GRF) data to deliver faster and detailed results in gait related applications [13]. These approaches are modifiable to effectively identify the gait abnormalities using spatiotemporal gait parameters. The present paper explores the performance of the deep neural network to classify parkinsonian and normal gait patterns.

A. Literature review

The sensitivity of the GRF data is crucial to classification of parkinsonian gait from normal gait. Pressure sensing is used to measure GRF generated in the foot during walking. A similar technique has been used to classify gait patterns among the children with Autism using ANN and support vector machine [14]. This paper has reported of technique to classify Autism gaits with a 95% of accuracy. This result has validated the effective use of machine learning approaches to classify pathological gait patterns. In [15], deep learning has been used with GRF data for wide area floor sensing. This study has explored the techniques of categorizing gait patterns by fusing raw spatiotemporal GRF. The paper has explored the use of CNN with long short-term memory approaches to study parkinsonian gaits. This technique has shown a 96% precision of identifying these gait patterns.

The technique introduced in [16] has used a computer vision-based technique to identify parkinsonian gait and a 95.49% accuracy has been reported. This study has been compared the normal gaits with parkinsonian gaits and has been able to successfully identify several progressive stages. Therefore, one of the primary goals of the present paper is to classify the parkinsonian gait based on the disease severity. Several previous attempts have used machine learning, vision-based systems, deep learning models, and statistical methods to classify parkinsonian gaits [14], [15], [16], [17], [18]. However, the classification of severity stages of parkinsonian gait is highly important for the treatments and have not been covered in many of these studies. Therefore, this paper has

focused on developing a cost effective and fast data processing application based on CNN to detect parkinsonian gaits to be apply for future medical applications.

This paper is structured as follows. Section 2 summarizes the database that is used for this study. Section 3 explains the methodology; data preprocessing, data matrix form development, image conversion, data preparation for CNN model, pattern recognition model development, and performance evaluation. Section 4 presents the CNN model performance and discusses the results and section 5 concludes the final outcome and presents recommendations for the future work.

II. Data Description

The data set used in this study was obtained from the open-source database PhysioNet [19]. The data included 93 idiopathic PD subjects and 73 healthy subjects with an average age of 66. This data consists PD disease severity measurement Hoehn and Yahr score for each subject. There were 10 subjects with the Hoehn and Yahr PD stage 3, 27 subjects with a Hoehn and Yahr PD stage 2.5, and 56 subjects with a Hoehn and Yahr PD stage 2. The dataset contains vertical GRF for both PD and healthy subjects with approximately 2 minutes of walking. The GRF was gathered using three separate measures, which are known as dual tasking (Ga), rhythmic auditory stimulation (Ju), and treadmill walking (Si) which is shown in Table 1. The dual tasking is used to evaluate the limitations of the attentional capability of the subjects [20]. The rhythmic auditory stimulation improves the gait and gait-related movements [21]. Similarly, the treadmill walking improves the gait and gait-mobility [22].

Table 1 Data description of the number of participants.

Group	Subjects	
	Number of PD subjects	Number of healthy subjects
Ga	29	18
Ju	29	26
Si	35	29

The data acquisition for every subject is carried out using eight flex sensors under each foot to capture the force applied to it during walking. Each sensor produced 100 samples per second and the readings are recorded in a raw matrix. After the data acquisition, all eight-sensor data from each foot is added and recorded separately into the matrix with their respective subjects [19].

III. Methodology

This section describes the data preprocessing, image conversion, development of the CNN and performance evaluation techniques.

A. Data preprocessing

The raw data files were converted from text to CSV for each sample and saved separately prior to the preprocessing. Each sample had 19 columns and data were recorded in a slightly different frame rate (some data had 12119 and others had 1000 frames per sample). The columns included were; time stamps, GRF data from the left foot of the each subject (eight sensor measurements), GRF data form the right foot of the each subject (eight sensor measurements), and the two total GRF values for each foot. The frame rate mismatch was handled as in [23], [24]. The final preprocessed data set after omitting the unnecessary data for CNN model, each subject data contained 18 columns and 500 frames. A single gait cycle including both heel strike and toe-off takes one second (100 frames), therefore, each subject data contained 5 gait cycles (i.e., GRF of each subject for 5 gait cycles). These samples were converted into a 500x18 matrices. The number of samples for each data class is as shown in the Table 2. The classified data for PD stage 2 and PD stage 3 had the maximum and the minimum number of samples respectively [25].

Table 2 Number of samples used for each classification.

Data class	**Number of samples**
Healthy subjects	2001
PD stage 2 subjects	2084
PD stage 2.5 subjects	1633
PD stage 3 subjects	541

B. Data Normalization

The data normalization is critical to deep learning models to map their inputs with the outputs. The differences in the scales must be omitted and data should be comparable with each other to corelate and identify their characteristics. This data set is normalized between 0 to 1 [26]. Each 500×18 matrix has considered independently in the data normalization process. After the normalization, the matrices are represented as an image with their respective color shades. The created spectrogram is contributed each with a height of 500 pixels and a width 18 pixels, totaling 9000 features. The reconstructed image samples are shown in Fig. 1.

The spectrograms represent the GRF of the data set; healthy subjects, PD stage 2 subjects, PD stage 2.5 subjects, and PD stage 3 subjects respectively from left to right. Similarly, 6259 images are created to represent all the data. The last two sections (columns) of the image are the total GRF of left and right foot respectively. Comparing to the samples of the healthy subjects, the PD subjects have shown a high total GRF due to their freezing of the gait during the walking.

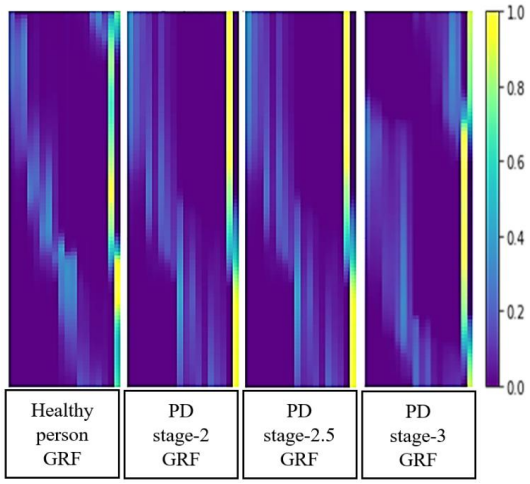


Figure 1 The normalized image samples. The yellow and purple colors represent the maximum and minimum GRF

C. Pattern Recognition Model Development

The data set is unstructured and labelled, therefore, supervised learning is suitable to be used in this application. Supervisory learning maps the data sets with inputs and outputs. The primary goal of this algorithm is to seek the relation between input images and GRF group labels. However, there is no quantifiable parametric variation between the gait patterns of PD and healthy subjects. Therefore, to identify the relation between GRF and PD, this proposed technique in this paper is used. Moreover, the developed CNN model can identify the PD stages concerning their Hoehn Yahr PD progress scales. The CNN model is used to learn the events and the phase changes during the gait cycles [27].

D. Convolution Neural Network (CNN)

CNN's capability of automatically grasping sophisticated patterns from large datasets is used for this application. In general, CNN consists of a convolution layer and a pooling layer [28], [29]. The images are initially fed into convolution layer which performs the convolution operation. The input convolution layer is set to 500×18×1 to fit with image size, including a single-color channel. This is a linear operation that multiplies set of weights with the input image's pixel values. The multiplication is achieved between an input pixel data array and two-dimensional weight array that is known as a filter [30]. The filters are used to determine the features that associate with each image. The convolution operation used in this algorithm is for a two- dimensional image and shown in the Equation 1 [31].

$$S(i,j) = (I \times K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \quad (4)$$

The parameters and variables are; two-dimensional image = I, two-dimensional filter = K, the indexes of rows and columns of the result matrix are marked with i and j respectively.

Table 3 Number of filters used for each layer

Layer description	Number of filters
Convolution layer	128
1st hidden convolution layer	256
2nd hidden convolution layer	512
3rd hidden convolution layer	1024

After applying the filter over a selected set of pixels, each value of the filter is multiplied with their corresponding value from the input image. Then all the multiplied pixels are summed up and placed in the output feature map. The features of the images such that edges and corners are identified using this convolution layer. Specifically, the generated output feature maps are maintained same size of the input data matrix by adding a padding value as a one into the convolution operation. And also, the rectifier linear function is applied to increase the non-linearity in feature map because the images have nonlinear features such as tractions between pixels, boarders, and colors. The model is constructed using four convolution layers with three hidden layers. The filter selection for each layer is given in Table 3. The filters are chosen concerning the maximum accuracy with the best performance characteristics [32].

The filters are selected in incremental order with the size of 3×3 to keep the feature space wide and shallow from the initial stages of the network while making it narrow and deep towards the end of the model. Every convolution layer is followed by pooling layer that reduces the spatial size of the representation [33]. As a result, the number of parameters and computation time in the network are minimized.

The pooling operation is carried out by sliding a two-dimensional filter over each channel of the feature map and then the features are summarized within the region that covered by the filter. After, pooling operation, the dimensions of the feature map $n_h \times n_w \times n_c$ are obtained using the Equation 2 [34].

$$y = \left(\frac{n_h + 2p - f}{s} + 1\right) \times \left(\frac{n_w + 2p - f}{s} + 1\right) \times n_c \quad (2)$$

Where, y is the dimension of output after the pooling operation, n_h is the height of feature map, n_w is the width of the feature map, n_c is the number of channels in the feature map, f is the size of the filter, s is the stride length, and p is the padding.

In this model, max pooling is selected by the pooling operation that selects the maximum element from the region of the feature map covered by the filter with the size of a 2×2 matrices [35]. The padding is selected to be zero by less considering the edges of the images. Similarly, to reduce the timed consumed for operation and complexity, the stride length is selected as 2. The convolution matrix of ith output after the max-pooling is a feature map containing the most outstanding features of the previous feature map. This is shown in Table 4. For instance, after the 1st pooling operation, the generated output feature map size is determined using Equation 2.

The height and width of feature map after the 1st convolution operation is 500×18. The other parameters are selected to be p=0, s=2, f=2, and n_c=1 (single color channel). By substituting the parameters into Equation 2, the generated output shape turned out to be 250×9×1. Similarly, each output feature map is produced sing the same technique.

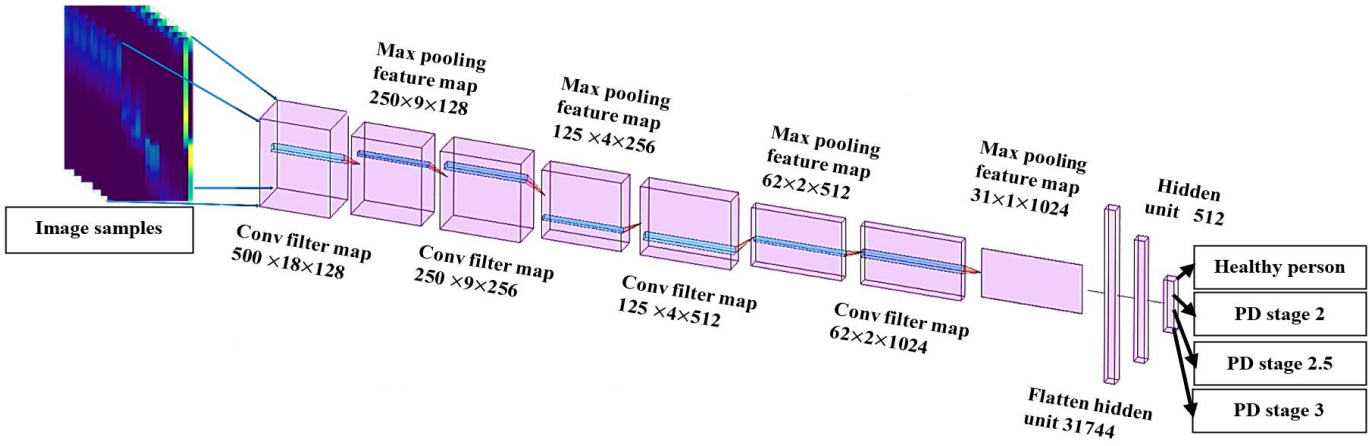


Figure 2 Illustration of the proposed CNN architecture

In general, overfitting provides too much information to CNN, specifically, by the information that is irrelevant to the model. Therefore, overfitting can be avoided by constructing a pooled feature map in the model [36]. After the pooled featured map is obtained, the fatten function is used to transform the entire pooled feature map matrix into a single column that could be fed into the neural network for classification. Flattening has resulted a single long feature vector that is used for the neural network. After flattening, the flattened feature map has been passed through another neural network. This step consists of an input layer, a fully connected layer, and an output layer. The fully connected layer is same as the hidden layer in ANNs. However, in this algorithm, it's fully connected. The output layer is the predicted class of each gait pattern [37], [48].

The identified pooled features are passed through the network and the error of prediction is calculated. The fully connected layer is developed using 512 neurons that consist of rectified linear unit (ReLU) activation function. Similarly, the output layer contains 4 neurons with softmax activation function that represents each output of the PD case's gait pattern. The softmax activation function is selected for the output layer to represent the probability of each class that brought down to numbers between zero and one. The model loss is calculated using categorical cross-entropy that could handle multi-class classification [39], [40].

The developed CNN model is optimized using an adaptive learning rate (Adam) optimization algorithm that is used to obtain individual learning rates of each parameter. The 80-percent of the dataset is used to train the model and the remaining is used to test and validate the model. The callback, early stopping functions are used to stop training when it reaches a certain accuracy with loss score by adjusting the learning rates over time and to prevent the overfitting during the training stage. Fig. 2 shows the proposed 2D-CNN architecture with the descriptions of each layer. The outcome of the model is determined by using human gait pattern analysis that specifies for each case [41]. CNN model is developed using open-source libraries that are found in SciKit learn, TensorFlow, and Keras (TensorFlow backend) [42].

E. Model Performance Evaluation

The performance of the classification algorithm is determined by the confusion matrix shown in Fig. 3. The gait pattern recognition of each subject is analyzed through the confusion matrix [43], [44].

Various performance measures are used to examine the model performance in the confusion matrix. Mainly, model accuracy, recall, precision, and F-measure parameters are obtained to evaluate the performance.

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Figure 3 Confusion matrix

The model accuracy is determined by Equation 3. Where true positive = TP, true negative = TN, false positive = FP and false negative = FN.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5)$$

Table 4 The output image shape for each pooling operation

Layer description	Output shape
1st pooling layer	250 × 9
2nd pooling layer	125 × 4
3rd pooling layer	62 × 2
4th pooling layer	31 × 1

The number of true positive value is determined using the recall function shown in Equation 4.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

F-measure is used to measure the model performance concerning precision and recall. Equation 5 shows the F-measure calculation formula [48].

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (5)$$

IV. Results

The results showed the prediction capability of the model as mentioned in the previous section. Moreover, model training and testing characteristics were visualized using graphs. The progress of the training, testing accuracy, and losses were plotted concerning 12 epochs as shown in Fig. 4 and Fig. 5 respectively. The model confusion matrix shows the comparison between prediction class and actual class resulted as in Fig. 6. The overall model performance scores are summarized as shown in Table 5, including each PD stage analysis.

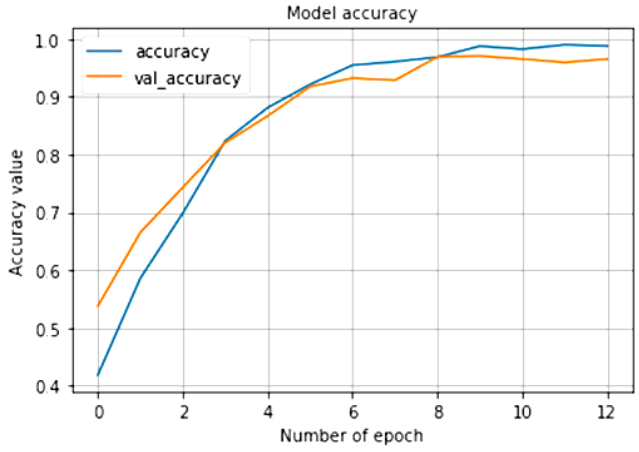


Figure 12 Model training accuracy and validation accuracy plot.

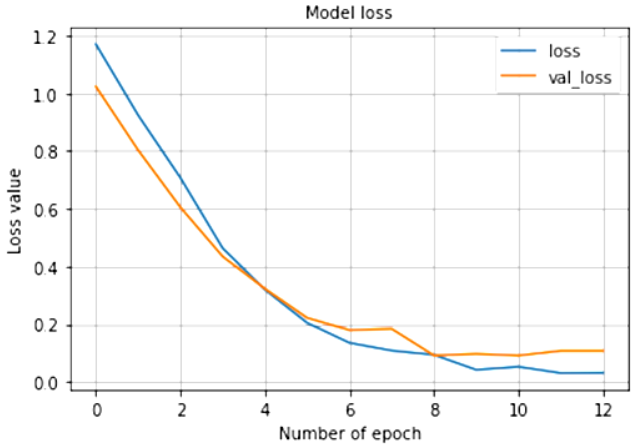


Figure 5 Model training loss and validation loss.

Table 5 Model performance.

Case	Precision (%)	Recall (%)	F1-measure (%)	Overall accuracy (%)
Healthy person	98	96	97	97
PD stage 2	93	97	95	97
PD stage 2.5	99	96	98	97
PD stage 3	97	97	97	97

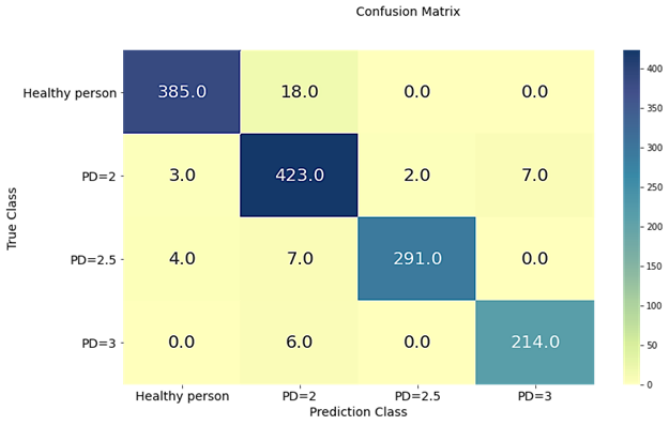


Figure 6 CNN model confusion matrix.

The results have shown that the proposed methods are capable of accurately predict the parkinsonian gait and its stages. Table 5 shows CNN algorithm performed with a maximum peak accuracy value of 97%. This technique improved the accuracy by 1% in comparison to the technique introduced in [15]. Although when compares to [15] the accuracy had increased by a 1%. The parameters; precision, recall, and F1-measure values were recorded over 93% accuracy showing the model reliability. This work has provided evidence that having a higher number of filters can enhance the model performance characteristics in this application. Fig. 4 shows the progress of the training and validation accuracy has reached 97% and confirmed that accuracy of the model. Similarly, the model training and validation losses were analyzed as shown in Fig. 5, achieving less than 0.2

obtaining a good fit. Moreover, the PD stage 2 showed the lowest precision value by scoring 93%. A key characteristic shown in results is the higher value in the super diagonal matrix. This is an indication of high accuracy of prediction in the CNN model. The current systems require only few apparatuses with respect to the vision-bases system in [16].

V. CONCLUSION

GRF data is a crucial parameter in human gait analysis to identify PD. This paper had presented a CNN based approach to identify and classify parkinsonian gait and stages of PD using GRF-based data recorded in human trials. The proposed technique had shown a 97% accuracy of prediction and the lowest precision recorded for classifying different stages of PD was 93%. Therefore, this work suggests CNN as a best approach to identify PD and their stages. The future work of this research would be helped for investigate other gait parameters of parkinsonian gait (e.g., joint velocities/acceleration, walking speed, etc.) for early diagnosing of PD.

REFERENCES

[1] A. Tsanas and S. Arora, "Large-scale Clustering of People Diagnosed with Parkinson's Disease using Acoustic Analysis of Sustained Vowels: Findings in the Parkinson's Voice Initiative Study," *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies*, 2020.

[2] M. Alhussein, "Monitoring Parkinson's Disease in Smart Cities," *IEEE Access*, vol. 5, pp. 19835–19841, 2017.

[3] E. Rovini, C. Maremmani, and F. Cavallo, "How Wearable Sensors Can Support Parkinson's Disease Diagnosis and Treatment: A Systematic Review," *Frontiers in Neuroscience*, vol. 11, 2017.

[4] L. Borzi, M. Varrecchia, S. Sibille, G. Olmo, C. A. Artusi, M. Fabbri, M. G. Rizzone, A. Romagnolo, M. Zibetti, and L. Lopiano, "Smartphone-Based Estimation of Item 3.8 of the MDS-UPDRS-III for Assessing Leg Agility in People With Parkinson's Disease," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 140–147, 2020.

[5] F. M. Skidmore, C. A. Mackman, B. Pav, L. M. Shulman, C. Garvan, R. F. Macko, and K. M. Heilman, *The Journal of Rehabilitation Research and Development*, vol. 45, no. 9, p. 1343, 2008.

[6] J. Kondragunta, C. Wiede, and G. Hirtz, "Gait analysis for early Parkinson's disease detection based on deep learning," *Current Directions in Biomedical Engineering*, vol. 5, no. 1, pp. 9–12, 2019.

[7] H. Fleyeh and J. Westin, "Extracting Body Landmarks from Videos for Parkinson Gait Analysis," *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, 2019.

[8] M. E. Morris, F. Huxham, J. Mcginley, K. Dodd, and R. Iansek, "The biomechanics and motor control of gait in Parkinson disease," *Clinical Biomechanics*, vol. 16, no. 6, pp. 459–470, 2001.

[9] A. Saboor, T. Kask, A. Kuusik, M. M. Alam, Y. L. Moullec, I. K. Niazi, A. Zoha, and R. Ahmad, "Latest Research Trends in Gait Analysis Using Wearable Sensors and Machine Learning: A Systematic Review," *IEEE Access*, vol. 8, pp. 167830–167864, 2020.

[10] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[11] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks," *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, 2016.

[12] A. K. Pandey, A. I. Khan, Y. B. Abushark, M. M. Alam, A. Agrawal, R. Kumar, and R. A. Khan, "Key Issues in Healthcare Data Integrity: Analysis and Recommendations," *IEEE Access*, vol. 8, pp. 40612–40628, 2020.

[13] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103678, 2020.

[14] N. M. Tahir and H. H. Manap, "Parkinson Disease Gait Classification based on Machine Learning Approach," *Journal of Applied Sciences*, vol. 12, no. 2, pp. 180–185, 2012.

[15] A. S. Alharthi, S. U. Yunas, and K. B. Ozanyan, "Deep Learning for Monitoring of Human Gait: A Review," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9575–9591, 2019.

[16] N. Kour, Sunanda, and S. Arora, "Computer-Vision Based Diagnosis of Parkinson's Disease via Gait: A Survey," *IEEE Access*, vol. 7, pp. 156620–156645, 2019.

[17] H. H. Manap, N. M. Tahir, and A. I. M. Yassin, "Statistical analysis of parkinson disease gait classification using Artificial Neural Network," *2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2011.

[18] N. Kour, Sunanda, and S. Arora, "Computer-Vision Based Diagnosis of Parkinson's Disease via Gait: A Survey," *IEEE Access*, vol. 7, pp. 156620–156645, 2019.

[19] Goldberger, A., Amaral, L., Glas, L., Hausdorff., Ivanov, P.C., Mark, R., …& Stanley, H. E. (2000). '*PhysioBank, PhysioToolkit, and PhysioNet'*: Components of a new research resource for complex phsiologic signals. Circulation [Online]. 101 (23), pp. e215-e220.

[20] S. O'shea, M. E. Morris, and R. Iansek, "Dual Task Interference During Gait in People With Parkinson Disease: Effects of Motor Versus Cognitive Secondary Tasks," *Physical Therapy*, vol. 82, no. 9, pp. 888–897, 2002.

[21] M. H. Thaut, R. R. Rice, T. B. Janzen, C. P. Hurt-Thaut, and G. C. Mcintosh, "Rhythmic auditory stimulation for reduction of falls in Parkinson's disease: a randomized controlled study," *Clinical Rehabilitation*, vol. 33, no. 1, pp. 34–43, 2018.

[22] P. Moyer, "Treadmill Exercise May Improve Parkinson Disease Symptoms," *Neurology Today*, vol. 6, no. 17, p. 20, 2006.

[23] A. Karatsidis, G. Bellusci, H. Schepers, M. D. Zee, M. Andersen, and P. Veltink, "Estimation of Ground Reaction Forces and Moments During Gait Using Only Inertial Motion Capture," *Sensors*, vol. 17, no. 12, p. 75, 2016.

[24] M. Sreenivasa, M. Millard, M. Felis, K. Mombaur, and S. I. Wolf, "Optimal Control Based Stiffness Identification of an Ankle-Foot Orthosis Using a Predictive Walking Model," *Frontiers in Computational Neuroscience*, vol. 11, 2017.

[25] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of Automatic Feature Extraction in Deep Learning Architecture," *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016.

[26] S. Ledesma, M.-A. Ibarra-Manzano, E. Cabal-Yepez, D.-L. Almanza-Ojeda, and J.-G. Avina-Cervantes, "Analysis of Data Sets With Learning Conflicts for Machine Learning," *IEEE Access*, vol. 6, pp. 45062–45070, 2018.

[27] S. Shinde, S. Prasad, Y. Saboo, R. Kaushick, J. Saini, P. K. Pal, and M. Ingalhalikar, "Predictive markers for Parkinson's disease using deep neural nets on neuromelanin sensitive MRI," *NeuroImage: Clinical*, vol. 22, p. 101748, 2019.

[28] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[29] D. Qin, J. Yu, G. Zou, R. Yong, Q. Zhao, and B. Zhang, "A Novel Combined Prediction Scheme Based on CNN and LSTM for Urban PM2.5 Concentration," *IEEE Access*, vol. 7, pp. 20050–20059, 2019.

[30] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019.

[31] Y. Tian, "Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm," *IEEE Access*, vol. 8, pp. 125731–125744, 2020.

[32] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep Learning-Based Classification of Hyperspectral Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[33] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[35] A. Ben-Cohen and H. Greenspan, "Liver lesion detection in CT using deep learning techniques," *Handbook of Medical Image Computing and Computer Assisted Intervention*, pp. 65–90, 2020.

[36] S. Wang, M. Zhou, Z. Liu, Z. Liu, D. Gu, Y. Zang, D. Dong, O. Gevaert, and J. Tian, "Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation," *Medical Image Analysis*, vol. 40, pp. 172–183, 2017.

[37] B. Ramsundar and R. B. Zadeh, TensorFlow for deep learning: from linear regression to reinforcement learning. Beijing: O'Reilly Media, 2018.

[38] S. U. Yunas, A. Alharthi, and K. B. Ozanyan, "Multi-modality sensor fusion for gait classification using deep learning," *2020 IEEE Sensors Applications Symposium (SAS)*, 2020.

[39] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

[40] A. Mehmood, M. Maqsood, M. Bashir, and Y. Shuyuan, "A Deep Siamese Convolution Neural Network for Multi-Class Classification of Alzheimer Disease," *Brain Sciences*, vol. 10, no. 2, p. 84, 2020.

[41] M. S. M. Guntor, R. Sahak, A. Zabidi, N. M. Tahir, I. M. Yassin, Z. I. Rizman, R. Baharom, and N. A. Wahab, "Convolutional Neural Network (CNN) based Gait Recognition System using Microsoft Kinect Skeleton Features," *International Journal of Engineering & Technology*, vol. 7, no. 4.11, p. 202, 2018.

[42] S. Mondal, K. Agarwal, and M. Rashid, "Deep Learning Approach for Automatic Classification of X-Ray Images using Convolutional Neural Network," *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019.

[43] S. Bhowmick, A. Nandy, P. Chakraborty, and G. C. Nandi, "Gait biometrics: A vision based approach for cloths invariant walking pattern classification," *2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC)*, 2013.

[44] J. L. Riskowski, "Gait and neuromuscular learning effects through the use of a gait monitoring system," *2009 IEEE International Conference on Rehabilitation Robotics*, 2009.

Predicting Student Performance in an ODL Environment: A Case Study Based on Microprocessor and Interface Course

H.R.S Senanayake [1] , A.P. Madurapperuma [2], G.S.N. Meedin [3]
[1,2,3] Department of Electrical & Computer Engineering, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka.
email : [1]savi3senanayake@gmail.com, [2]apmad@ou.ac.lk, [3]gsnad@ou.ac.lk

***Abstract*— The Open University of Sri Lanka adopts an Open and Distance Learning philosophy that breaks down many barriers for students to pursue their higher education dreams. One critical feature of Distance Learning is the onus on the students to be in charge of their learning. Academics and academic support staff facilitate this learning process by conducting many activities. But by the end of the course period the performance of students varies where some students get classes and others fail. The university has taken several steps to support learners while they are pursuing their degree programs, but the effectiveness of these initiatives is not as good as desired. In this project we propose a system of continuous monitoring of student's academic activities to build a dynamic profile of the student throughout the study program. The proposed system will analyze student activities continuously and will predict the future performance based on the student's own data as well as patterns identified by analyzing the data generated by previous students. The system will provide this information to the academic and academic support staff as well as the student so corrective action can be taken when a student is expected to perform badly in future activities. The system uses an Artificial Neural Network (ANN) model for predicting the performance of the student and this model can predict the students' performance in accuracy of 93.7.**

Keywords*— *Artificial Neural Network (ANN), Learning Analytics, Deep Learning, Learner Support

I. INTRODUCTION

The Open University of Sri Lanka uses Open and Distance Learning (ODL) methodology to conduct its academic programs. ODL approach removes barriers to open education and provides flexible learning opportunities to anyone, anywhere, any time and at any pace. It has been using a practice that is recently named, the "flipped classroom model" since its inception. Students are provided learning material at registration that they study on their own. Their learning process is supported through a series of day schools, lab classes, field trips and other face to face activities. Their progress is continuously evaluated through Tutor Marked Assignments (TMAs), LAB evaluations, Continuous Assessment Tests (CATs), Mini projects, viva and other forms of formative evaluation often culminating in a summative evaluation in the form of a final examination or a final viva. In an ODL system, the distance between the learners and the facilitators is also a challenge that needs to be addressed. Due to the distance there is a lack of opportunity for facilitators to observe the academic progress of the students continuously and make necessary course corrections to ensure academic success. This is due to the lack of a proper system to continuously monitor and analyze student performances, thus providing the academics an opportunity to intervene to improve the situation based on actual data. Only a few milestone assessments are carried out and the overall performance is often measured through a final examination. This creates a situation which leads to low completion rates, because by the time the results of the examinations are known it is too late to take any action to improve the situation. It is known that continuous motivation is essential for a student's success in an ODL environment. Often, the students whose performance is low in formative assessments dropout of their studies taking the easy way out. Thus, it is important for an institute working on ODL mode to devise a mechanism to continuously monitor the student performance and carry out remedial actions when the student is not faring or expected to fare well in their studies. Therefore, we need a better solution to increase the output of the university.

In this paper we describe a system for predicting student performance using deep learning technology. Using this system, a student's activities are regularly monitored and evaluated based on the current values of a set of identified parameters as well as historic data. The output is a prediction of the students' performance in the future. This data will be useful to the students as well as academics to make necessary changes to increase the chances of academic success. We can analyze the performance of the student by considering the various factors of the students. Some of these factors include attendance of the lectures, marks of the TMAs, marks of the CATs, marks of the presentation, and marks of the Labs etc. University could identify the factors for low performance earlier and is able to predict students' behavior. This knowledge can help them in taking proactive actions, to improve the performance of such students. Students will be able to identify their weaknesses beforehand and can improve themselves. Lecturers will be able to plan their lectures as per the needs of students and can provide better guidance to such students.

II. LITERATURE REVIEW

Many systems were implemented in the industry to perform student profiling and data mining. Some of them are just student management systems and others are predicting/ data mining systems. This section summarizes the different works that have been done related to this area.

S. Naser and his colleagues published a paper in the title of "Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology". Their work describes the

predicting performance of a sophomore student enrolled in engineering majors. A number of factors have been identified that may possibly influence the performance of a student. The aims of this study are identify some suitable factors that affect a student performance, convert these factors into forms appropriate for an adaptive system coding and model an Artificial Neural Network that can be used to predict a student performance based on some predetermined data for a given student. Input variables have been identified which can simply be obtained from the student file and registrar system. The input variables are High School Score, Results in Math I in the student freshman year, Results in Math II in the student freshman year, Results in Electrical Circuits in the student freshman year, Results in Electronics I in the student freshman year, CGPA of the freshman year, Type of high school whether it is in Palestine or outside and Student gender [1]

Thakar and Mehta, try to understand the behavior of the students, and try to solve the problem of student turnover and unemployed students in their paper "Performance Analysis and Prediction in Educational Data Mining [2]. This is a survey paper. They have summarized many paper details. Their stakeholders are universities/institute, management, teachers, students, parents. They analyze and predict with the help of data mining techniques. Such as Neural Network, Decision tree, K-means clustering, Association Rule Mining.

Thakar and Mehta work (2018) describing the Predicting students' results in higher education using a neural network**.** They were observed the poor quality of graduates of some Nigerian Universities in recent times. It has been partly traced to inadequacies of the National University Admission Examination System. In this study an Artificial Neural Network (ANN) model, for predicting the likely performance of a candidate being considered for admission into the university was developed and tested. Various factors that may likely influence the performance of a student were identified. Such factors used as input variables for the ANN model. Their model uses a Neural Network with one input layer, two hidden layers and one output layer. The following factors were used as input features. Type of study program (distance education /part time or full time education), Gender of the student, High-School graduation GPA, Age of the student and Difference in years from the moment the student graduates high-school until he/ she enrolls at university. They classified students according to students GPA after the first year of study in three classes:

POOR RESULTS (those students with GPA lower than 6), MEDIUM RESULTS (those students with GPA between 6 and 8) and GOOD RESULTS (those students with GPA greater than 8). [3]

P Jain and his group describe their study in an Artificial Intelligence-Based Student Learning Evaluation conference paper. They aim to discover the knowledge for analysis student motivation behaviour on e-Learning based on Data Mining Techniques, in case of the Information Technology for Communication and Learning Course at Suan Sunandha Rajabhat University. The data mining techniques were applied in this research including association rules, classification techniques.[4]

D. Kabakchieva was studied Mining Educational Data to Analyze the Student Motivation Behavior. His study presents the results from data mining research, performed at one of the famous and prestigious Bulgarian universities, with the main goal to reveal the high potential of data mining applications for university management and to contribute to more efficient university enrolment campaigns and to attract the most desirable students. The research is focused on the development of data mining models for predicting student performance, based on their personal, pre-university and university-performance characteristics. The dataset used for the research purposes includes data about students admitted to the university in three consecutive years. [5]

III. Artificial Neural Network

An Artificial Neural Network (ANN) is a mathematical model that tries to simulate the structure and functionalities of biological neural networks. Basic building block of every artificial neural network is an artificial neuron, which is a simple mathematical model. The brain is a highly complex, nonlinear and parallel information processing system that has the capability of organizing neurons so as to perform certain computations. It is many times faster than the fastest digital computer. A neural network contains an interrelated set of artificial neurons and it processes information using a connectionist form to computation. A Neural Network can model and it can train a model in nonlinear and very complex relationships which is most important because the relationship between inputs and outputs are nonlinear as well very complex. After learning from the initial inputs and their relationships, it can infer unseen relationships on unseen data as well. And it does not impose any restriction on the input features. We can use data with high volatility.

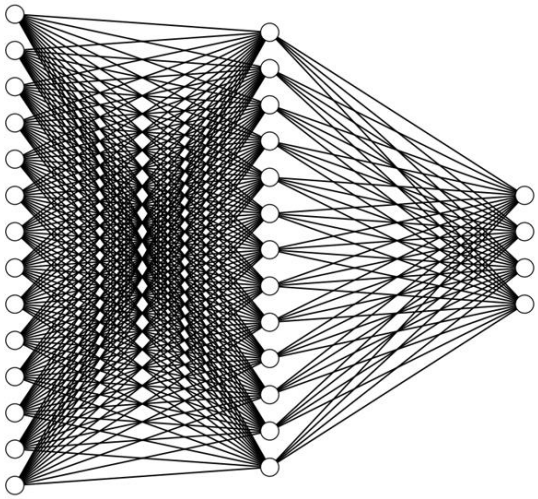


Figure 1 Neural Network Diagram

An Artificial Neural Network emulates the human brain in solving a problem, is a more common approach that can tackle this kind of problem. A Neural Network will take the input data and push them into an ensemble of layers. The network needs to evaluate its performance with a loss function. The loss function gives to the network an idea of the path it needs to take before it masters the

knowledge. The network needs to improve its knowledge with the help of an optimizer.

A. Input Variable

In order to model student performances it is important to identify the correct parameters that will provide an insight into the students' performance. The following were used to identify a set of parameters as input to our model.

Studying the research papers which are most related to this topic area and the technologies

Distributing the questionnaire among the students who are currently following the degree program

Studying of the academic model used by the Open University of Sri Lanka.

IN order to test our hypothesis, the model was developed for the Microprocessor and Interfaces course. This course is done by many students who follow BTech Engineering degree program as a compulsory course. It was also noted that the number of students who get eligible for the final examination as well as those who pass the exam is comparatively less than other similar courses. Thus, the course presented itself as an ideal choice where a prediction on the student performances can make a great difference in the success rate of the students. We selected 14 genetic and non-genetic factors according to the particular course structure. A description of the importance of each factor is described at continuation.

Open University consists of an ODL system. Therefore, the students who are following the degree program have to put a huge effort to study. In the "flipped classroom model", students study the learning material at their leisure, hence attending the Day Schools and the LAB session takes a very important role. Because they can discuss the concepts they have learnt during their studies directly with the lecturer and understand how these concepts are applied in practice under the guidance of the lecturer. Lab classes allow them to have hands-on experience. These two activities have a big impact in their learning process. TMA, LAB, CAT marks measure the students' performance during the course. These are formative assessments that allow the student and the academic conducting the course to have an idea how the student is progressing in his academic work. A student use of Library facilities gives an indication of the learning process of the student as it indicates that the students are delving deeper into the subject matter. Using the university's student portal named MyOUSL and ELearn facilities students share knowledge with peers and they can interact asynchronously with the lecturer to clarify their doubles or have a meaningful discussion on the material being studied. Participation in activities organized by different societies and in sports events is an indication of the wholesome experience that the student is enjoying at the university. Further, often these activities are organized with sponsorships from companies giving the students a first-hand opportunity to interact with representatives of these companies and become aware of new trends and technologies. Studies have shown that exercise increases blood flow to the brain and helps the body build more connections between nerves, leading to increased concentration, enhanced memory, stimulated creativity and better developed problem solving skills [11]. In short, playing sports helps everyone's brain grow and makes it work better. We have used a model to study how these factors affect the students' performance.

In summary, we have identified the following features as inputs to the model.

1. Day School Attendance
2. Lab Attendance
3. TMA1 Marks
4. TMA2 Marks
5. TMA3 Marks
6. LAB1 Marks
7. LAB2 Marks
8. LAB3 Marks
9. CAT Marks
10. Using Library Facilities
11. Using My OUSL Facilities
12. Using ELearn Facilities
13. Interaction with Society
14. Participation of Sport

Above input factors were converted into a format that can be used in the Neural Network Analysis. Table 01 shows the transformation for all 14 input features.

Table 1 Transformation of input data

No	Input Variable	Domain	
01	Day School Attendance	1 2 3 None (No of days)	1 2 3 4
02	Lab Attendance	1 2 3 None (No of days)	1 2 3 4
03	TMA1 Marks	Above 80% 60 – 79 40 – 59 0 – 40	1 2 3 4
04	TMA2 Marks		
05	TMA3 Marks		
06	LAB1 Marks		
07	LAB2 Marks		
08	LAB3 Marks		
09	CAT Marks		
10	Using Library Facilities (per week)	More than 20 h 15 – 20 07 – 15 0 – 07	1 2 3 4
11	Using My OUSL Facilities (per week)		
12	Using ELearn Facilities (per week)		
13	Interaction with Society (per week)		
14	Participation of Sports (per week)		

B. Output Variable

In our model, the output variable represents the student's academic performance. Output variables were categorized into four categories as “Excellent”, “Good”, “Moderate” and “Weak”.

IV. METHODOLOGY

Artificial Neural Network is an algorithm that uses past data to train a model. It can learn from the training data. Using the ANN trained using historical data, it is possible to produce an output which is the predicted performance of a new student. Neural Network can model and train a model in non-linear and in very complex relationships. And it does not impose any restriction on the input variables. Therefore, we can decide the number of inputs and features according to our study. Further Neural Network can model with high volatility. Our Model uses three layers as follows: 14 neurons are used in the input layer and 13 neurons in the hidden layer with and 4 output neurons in the output layer. The Neural Network activation function is responsible for transforming the summed weighted input from the nodes into the output for that input. “ReLU” function was used as activation function of input layer and hidden layer and “Softmax” function was used as activation function of output layer. Inputs are processed using the above functions to produce the desired output. "ReLU" function can converge network very quickly and the “Softmax” activation function is used for the output layer and when we categorize our inputs into multiple categories.

The model also used feed forward back propagation algorithms for training the data set. ”Adam” is used as an optimizer. Because it shapes and mold our model into its most accurate possible form by futzing with the weights.

A. Dataset

Since this is a faculty course most of the students are registered for these selected courses. 1068 records from the last three years student’s data were used for the purpose of training and testing. 907 samples were used to train the model and 161 samples were used to validate the model.

B. Implementation

The system was developed as a web based system. Frontend of the system was created using PHP, HTML and all data stored in the MySQL database. A GUI was built to collect student information as well as to display their profiles and to display their performance level. “TensorFlow” framework was used to create the neural network model. Python is used as the language to build a neural network.

The Model was created and trained using the historical data (2016 - 2018). Once the model was trained, new students can enter their data to the system to view predicted performance as the output. When a student enters the data, the output is also communicated to the academic in charge of the course. The process following to build the ANN system is shown in Fig 2.

C. Technologies Used to build the system

TensorFlow: TensorFlow is an end to end open source platform for machine learning. It is a free and open source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. [7]

Pandas: This is a software library written for the python programming language for data manipulation and analysis. Pandas library is supported in the TensorFlow environment. It is a data structure and data analysis tool for python programming. [8]

Keras: Keras is an open source neural network library written in python programming language. It is capable of running on the TensorFlow. It is designed to enable fast experimentation in neural networks. [9]

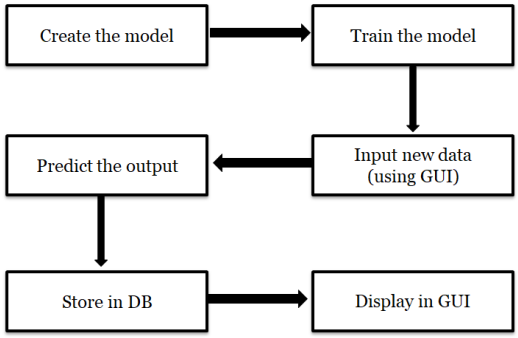


Figure 2 Describe the system overview

V. RESULT AND DISCUSSION

As explained in section 4, the model used in this work consists of three layers. Input layer has 14 neurons, the hidden layer has 13 neurons and the output layer has 4 neurons. Input layer and hidden layer use the "ReLU" activation function and output layer uses "Softmax" activation function. Fig 3 shows the script used to configure the model.

```
#model configuration.
model = Sequential()
model.add(Dense(14, activation='relu',  kernel_regularizer=regularizers.l2(0.001),input_shape = (14,)))
model.add(Dense(13, kernel_regularizer=regularizers.l2(0.001), activation='relu'))
model.add(Dense(4,  kernel_regularizer=regularizers.l2(0.001),activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history  = model.fit(x_train,y_train, epochs = 200, batch_size = 5, validation_data = (x_test,y_test))
model.save('Model_A.h5')
```

Figure 3 Script of Configuration

Fig 4 shows the number of interconnections between different layers of the model. It consists of dense layers which are the deeply connected neural network layers that is the most common and frequently used layer. It performs the following operations on the input and return output.

output = activation(dot(input, kernel) + bias) (1)

input - input data

activation - activation function

dot - numpy dot product of all inputs and its corresponding weights

kernel - weight data

bias - biased value in machine learning to optimize the model

In the model, the summary displays the following information.

The layers and their order in the model.

The output shape of each layer.

The number of parameters (weights) in each layer.

The total number of parameters (weights) in the model

Using the "Keras Library" we can show whether neural network parameters are trained or not. In order to calculate

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 14)                210
_________________________________________________________________
dense_2 (Dense)              (None, 13)                195
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 56
=================================================================
Total params: 461
Trainable params: 461
Non-trainable params: 0
```

Fig. 4. Summary output of the model

the number of parameters within an individual layer we need the following data:

Number of inputs to the layer

Number of output from the layer

Whether or not the layer contains biases

The output is calculated as

$$Z = XiWi + b \quad (2)$$

\Where

Xi - Value of input parameters

Wi - Weight of the parameters

b - bias

To calculate the number of parameters, the number of inputs to the layer is multiplied by the number of outputs from the layer and the number of biases in the layer added to the result. According to the calculation, input layer has 210 (14*14+14) parameters. Hidden layer has 195 (14*13+13) parameters. And output layer has 56 (13*4+4) parameters. This model has 461 total parameters. We used the "model.fit()" method of Keras library to train the parameters using historical data as shown in Fig 3. The summary output produced by the "model.summary()" method as shown in Fig4. In Fig4 dense_1, dense_2, dense_3 are input, hidden, output layers respectively. In the code output shape (None, 14) indicates whether we have a pre-defined number or not and corresponding output to the next layer.

We use a confusion matrix to lay out how many predicted categories or classes were correctly predicted and how many were not. It allows the visualization of the performance of an algorithm. It produces predicted values in this case, the student's future performance only by looking at the input value as a matrix. The X axis represents the "Predicted Values" and the Y axis represents the "True Values". Diagonal describes the values which model has predicted correctly. Values outside the diagonal represent the values which were predicted incorrectly. When we run our model the system produces the confusion matrix in Jupyter Notebook using Tensorflow Environment. In this model, the Confusion matrix used 161 test data from the total data set of 1068 values as described in section 4.1. Among the 161 test data 151 data were predicted correctly. Correctly predicted values are displayed along the diagonal. Ten values are outliers

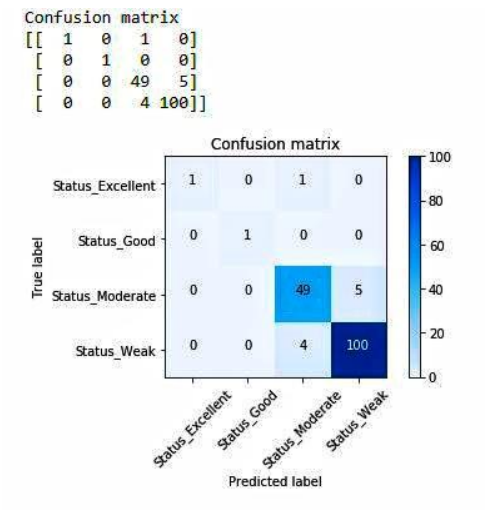


Figure 5 Confusion matrix for the model

Fig 6 represents the validation and training accuracy of the model. Y – Axis represent the accuracy level and the X – Axis represent the number of Epochs. Training accuracy is the fraction of predictions the model got correct using training data. Validation accuracy is the fraction of prediction the model got correct using testing data. Training accuracy is displayed using the "dotted line" (blue color) and validation accuracy is displayed using the "line" (red color). 200 epochs were used to the system. It displays the variation of the accuracy for each epoch. As shown in Fig 6 in between 0- 25 epochs, training accuracy and validation accuracy increased very quickly. Between 25 - 50 epochs, accuracy slowly increased. Then up to 200 epochs accuracy remains steady with a little variation around 93% accuracy level. Thus, the graph shows that beyond 50 epochs the model displays a high accuracy in both training and validation. We can consider that the model can be used to predict student performance with a high accuracy.

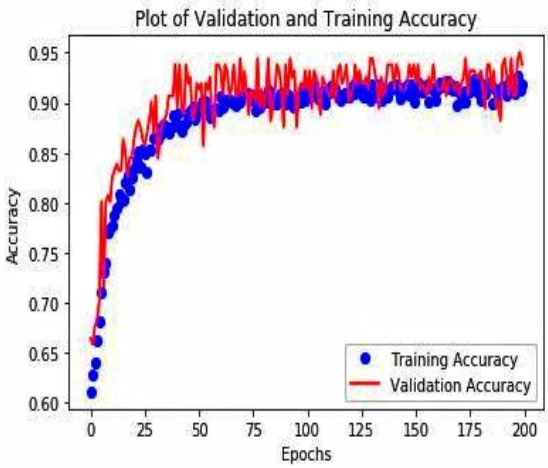


Figure 6 Plot of the Validation and Training Loss

Fig 7 represents the validation and training loss. Y – Axis represent the loss level and the X – Axis represent the number of Epochs. Training loss is nothing but a prediction error of a neural network using training data. Validation loss is a number indicating how bad the model prediction using test data. Training loss is displayed using the "dotted line" (blue color) and validation loss is displayed using the "line" (red color). Between 0 - 25 epochs, training and validation loss rapidly decreases. Between 25 - 75 epochs, loss reduces slowly. Up to 200 epochs, loss decreases very slowly. According to this graph model loss is very low. This collaborates that the accuracy of prediction using our model is high.

Samples of testing dataset which are used as model inputs are shown in Fig 8 and respective outputs which are extracted from the test dataset are shown in Fig 9. These are randomly selected data from the test dataset. Fig 10 shows the results which are predicted by the model for corresponding inputs which are shown in Fig 8. When comparing Fig 9 and Fig 10 it clearly shows the success of this model. Because actual outputs and the predicted outputs are the same for the used input factors. It is proof that the model can predict the student performance in a very accurate manner.

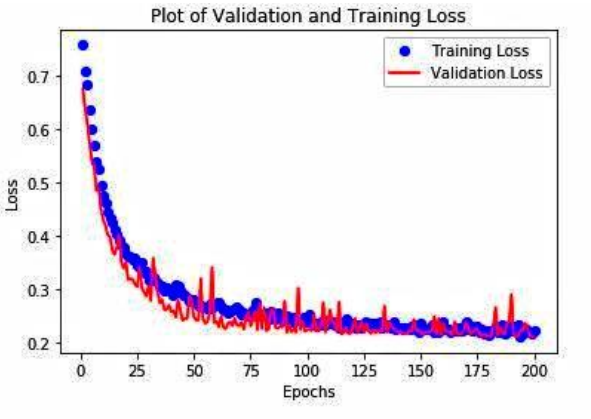


Figure 7 Plot of the Validation and Training Accuracy

\

	Day_Scl_Attendance	Lab_Attendance	TMA1	TMA2	TMA3	LAB1	LAB2	LAB3	CAT	Library_Facilities	My_OUSL_Login	ELearn_Login	Society	Sports
260	4	4	4	4	4	4	4	4	4	4	4	4	4	4
569	4	3	2	4	4	3	3	3	4	3	3	3	3	3
529	4	4	4	4	4	4	4	4	4	4	4	4	4	4
10	3	2	2	2	3	2	2	2	4	2	2	2	2	2

Figure 8 Sample input data from test dataset

	Grade_Excellent	Grade_Good	Grade_Moderate	Grade_Weak
260	0	0	0	1
569	0	0	1	0
529	0	0	0	1
10	0	1	0	0
111	0	0	1	0

Fig. 9. Sample output data from test dataset

```
260  Student Input:= [[4 4 4 4 4 4 4 4 4 4 4 4 4 4]]
Student Status:= ['Weak']
569  Student Input:= [[4 3 2 4 4 3 3 3 4 3 3 3 3 3]]
Student Status:= ['Moderate']
529  Student Input:= [[4 4 4 4 4 4 4 4 4 4 4 4 4 4]]
Student Status:= ['Weak']
10  Student Input:= [[3 2 2 2 3 2 2 2 4 2 2 2 2 2]]
Student Status:= ['Good']
111  Student Input:= [[4 3 3 3 4 3 3 3 4 3 3 3 3 3]]
Student Status:= ['Moderate']
```

Fig. 10. Predicted results by the model

VI. Conclusion

In this paper we have described a system based on Artificial Neural Networks to predict students' performance during a course. 14 parameters that are considered as important for a students' success in a course are used as the inputs and depending on the values future performance of a student was predicted. First of all we gathered student details. Then preprocessed the data to identify the data set consisting of any non-unique values and non-applicable values. Then the model was created by using the Neural Network Algorithm. Data set was divided into two parts as training data and testing data. Model was trained using training data and tested by using test data. Model summary was generated by using the "Keras". Then the confusion matrix was plotted for better visualization of the test data behavior. Two important graphs were plotted to display the accuracy and the loss when we train and test the model using training data and the testing data. Using these two graphs we can clearly see how this model varies for each epoch. According to the model summary, confusion matrix and the accuracy and loss graph clearly display the success of our model. In mathematically we can say that this model can predict the students' performance in accuracy of 93.7.

We do not use pre-enrollment data in the system, but have identified that they can have a strong relationship to how a student will perform in the course. The next version of the system will incorporate pre-enrollment parameters such as the demography, social status, performance at school level exams, and language proficiency as input parameters. The system can be used by the students as well as academics in charge of the courses to assess a student's predicted performance. This would provide an opportunity for both the student and the academic to take corrective action if the student is expected to perform badly. Through this system, students with academic problems can be identified early and course corrections applied to change their performance from weak to excellent.

In conclusion we have built a system that can improve the success rates of students following study programs in Open and Distance Learning mode through continuous monitoring and predicting future performance. The system can be used by students themselves and academic staff to improve completion rates for ODL students based on the predicted values we plan to link the system to the universities managed information system. Evaluation can be done automatically. Once done on a continuous activity throughout the student's academic career

References

[1] Naser, S., Zaqout, I., Ghosh, M., Atallah, R. and Alajrami, E.: Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology. International Journal of Hybrid Information Technology, 8(2), pp.221-228(2015)

[2] Thakar, P., Mehta, A.: Performance Analysis and Prediction in Educational Data Mining: A Research Travelogue.International Journal of Computer Applications (0975 – 8887),10(15) (2015)

[3] Oancea,B. , Dragoescu, R., Ciucu, S.: Predicting students' results in higher education using a neural network.International Conference on Applied Information and Communication Technologies (AICT2013), pp. 25.-26.Jelgava, Latvia (2013)

[4] Jain, G., Gurupur, V., Schroeder, J. and Faulkenberry, E.: Artificial Intelligence-Based Student Learning Evaluation: A Concept Map-Based Approach for Analyzing a Student's Understanding of a Topic. *IEEE Transactions on Learning Technologies*, 7(3), pp.267-279(2014)

[5] Kabakchieva, D.: Predicting Student Performance by Using Data Mining Methods for Classification.

Cybernetics and Information Technologies, 13(1), pp.61-72(2013)

[6] .Machine Learning Mastery, machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=In%20a%20neural%20network%2C%20the

[7] Wikipedia ,https://en.wikipedia.org/wiki/TensorFlow#TensorFlow

[8] Wikipedia, https://en.wikipedia.org/wiki/Pandas_(software)

[9] Wikipedia, https://en.wikipedia.org/wiki/Keras

[10] Calvert, C.: Developing a model and applications for probabilities of student success: a case study of predictive analytics. *Open Learning: The Journal of Open, Distance and e-Learning*, 29(2), pp.160-173(2014)

[11] Tomporowski, P. D., & Pesce, C. (2019). Exercise, sports, and performance arts benefit cognition via a common process. Psychological Bulletin, 145(9), 929–951. https://doi.org/10.1037/bul0000200

Naive Bayes Based Approach on Telecommunication Churn Prediction Using Financial Data

Buddhika Livera [1], D D M Ranasinghe [2]
[1,2] Department of Electrical and Computer Engineering, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka.
email: [1] wbslivera@slt.com.lk, [2] ddran@ou.ac.lk

***Abstract*—In the telecommunication industry, identifying customers who tend to leave as early as possible has become a mandatory task for survival. The motive behind the early detection of churn is that with the oversaturated market, it costs more resources to gain a new market share than to recover a lost market share. Therefore, the most effective approach to sustain is the prevention of churn. In general, in the telecom industry revenue generation system is sepa-rated from the management systems of the company. However, through the revenue realization process, finance data received gives some insight into customer data. This study shows how those insights could be applied with a Naive Bayes based approach to predict churn with the accuracy of over 85%. As the suggested approach is based on financial data, it enables the integration with management systems such as ERP compared to others which are based on consumer usage**

Keywords— Telecom Churn, ERP, Naïve Bayes, Fintech

I. INTRODUCTION

At present, there are many telecommunication service providers, and one single service provider may not be able to maintain the market share as the most popular among the customers throughout the timeline. It is very often that people express dissatisfaction with some aspect of their service provider, it could be either quality issues, unwanted product activations, poor customer service, unfair pricing plans, or confusing billing schedules. Due to the business nature, the industry is more vulnerable to churn. Also, it is not only the frustration that causes churn, in an over penetrated market availability of options also contributes to churn.

The churn itself is complex and a challenging problem because it causes more ad-verse effects on the service providers in terms of effort and cost required to attract new customers than to regain the left customers [1]. Due to this very reason, all most all the leading companies are spending a considerable amount of resources to retain their customer base to achieve a consistent customer base.

There have been many efforts and research carried out to identify the potential of churn with various success rates. Though there are many prediction models with different approaches such as Yu Zhao, Bing Li with One-Class Support Vector Machine [2], and NNA Sjarif, NF Azmi with Multilayer Perceptron [3], the main source of data is based on the consumer usages and patterns generated by revenue generation (usually the billing system data) system. Because of the decoupling nature between management systems and billing systems in the business domain, relating prediction outcomes with other systems become complex or an impossible task. Instead of using consumer usage data, this paper proposes an approach based on financial information generated during a typical revenue realization process to predict the churn.

A Typical Telecommunication Business Model

In a modern telecommunication business setup, there is a billing system that drives core revenue generation with other facilitating systems such as Customer Relation-ship Management (CRM), Operation Support (OSS), and cashiering. However, to manage supportive business operations, such as finance, human resources, procurement, and inventory-related transactions, it is common to use Enterprise Resources Planning (ERP) systems or similar, in place by integrating with other systems [4]. One such typical setup is presented in figure 1, which is the setup of one of the leading telecommunication service providers in Sri Lanka.

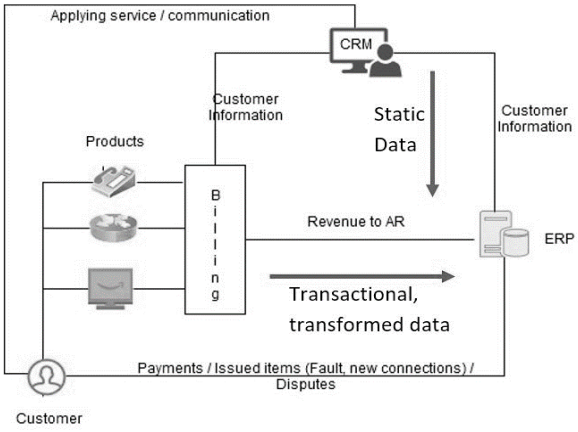


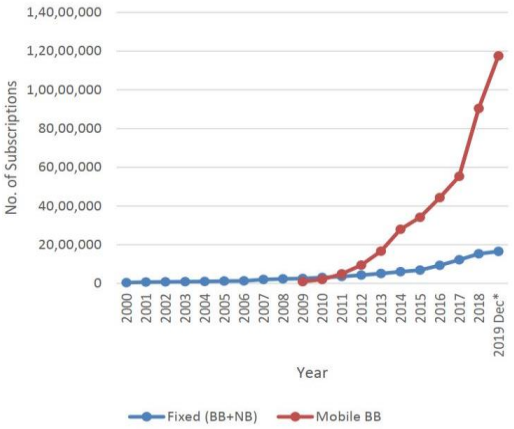
Fig. 13. Set up of a modern telecommunication service provider

Set up of a modern telecommunication service provider

As illustrated in the figure. 1, there are various activities related to different systems. Yet, some insights of all most all the activities can be obtained by analyzing the ERP management system.

A. Business Environment

This study was carried out in a rapidly changing business environment with many opportunities for growth due to the recent overwhelming extension of the ICT sector of the country, during the last decade. The statistics of the following figure 2, from the Telecommunications Regulatory Commission of the country indicates the growth of the telecommunication industry during the past two decades [5].



BB – Broadband / NB – Non-Broadband

Fig. 2. Fixed & Mobile Broadband Growth (2000 – 2019)

The approach and the model discussed in this study are based on historical data obtained from an ERP system of a leading telecommunication service provider in the south Asian region located in Sri Lanka. As per the company profile presented in its annual report of 2019, it has a market share of 1.4 million fixed voice, 0.5 million IPTV, and 1 million fixed broadband connections [6].

This paper contains four main sections describing the various aspect of the study. In the subsequent section, the related work is discussed to elaborate theoretical grounds, followed by the methodology explaining the data and suggested approaches to realize a prediction model. The paper is concluded with the results and discussion describing the findings and future work.

II. Related Work

Ever since the problem of telecommunication churn appears, there have been many efforts to identify churn on an individual and organizational scale. In the published paper by Dahiya, K., and Talwar K, reviews and summarizes the most common approaches that could achieve high accuracy rates on predicting churn [7]. In their study, they concluded that the decision tree-based techniques, Neural Network-based techniques, and regression techniques are generally applied in predicting customer churn in the telecommunication domain. It also highlights how decision tree-based techniques outperformed some of the existing data mining techniques such as regression in terms of accuracy while the neural networks exceed the performance of decision tree-based techniques due to the size of datasets used and the capability of different feature selection methods.

A group of researchers has also done a similar evaluation of various models on sub-scriber dissatisfaction and improving retention in the wireless telecommunications industry [8]. In their paper, they categorized features from a dataset of 46,000 primary business subscribers into five categories, network, billing, application for service, Market, and demographics. They noticed that the features under the market could be ignored when the prediction is made for shorter periods of intervals as for a shorter period market remain the same. As the outcome of the research, the paper claims a reduction in the churn of 40% because of early identification and treatment over a period of six weeks. Following table 1, summarize the categorization of features they identified and a point to highlight is that like many other efforts, the source of the features is the billing system.

Though there are many proposals to predict churn, based on the datasets, the different model tends to result in various success rates. For instance, in the research Simsek Gursoy, Tugba [9] has compared regression techniques with decision tree-based techniques and found that in logistic regression, the accuracy of the analysis churn prediction is 66%, wherein the case of decision trees the accuracy measured is 71.76%. However, in the paper published by Saini, Nisha, Monika, and Garg, Kanwal claim that with the decision tree-based approach, they could achieve 90% accuracy [10].

In the area of focused revenue and customer behavior in the telecommunication domain, few papers highlight the effective application of Bayesian networks. A re-search paper published by Geng Cui and M Leung Wong [11] proposes a Bayesian network model together with evolutionary programming for effective direct marking in the same domain.

Many organizations increasingly move towards ERP systems because it unites the entire functions of the company by maintaining many transactions under a single system. Due to this very nature of the ERP systems, there has been much interest in deriving intelligence from ERP systems. According to the findings by Rouhani, Saeed & Mehri, M [12] proposes that, ERP implementation promotes information-based decision making of an organization that enhances the organizations' business

Table 1. Factor Importance and Nature of data required for prediction

Factor	Importance	Nature of data
call quality	21%	network
pricing options	18%	market, billing
corporate capability	17%	market, customer service
customer service	17%	customer service
credibility / customer communications	10%	market, customer service
roaming / coverage	7%	network
handset	4%	application
billing	3%	billing

intelligence readiness.

Based on the above presented research work, it is possible to conclude that majority of efforts on predicting telecommunication churn depends on features extracted from consumer usage data. As a result of the nature of data that is mostly numerical, logistic regression-based techniques have become the preferred techniques over other techniques for predicting churn in the telecommunication industry. Finally, an interesting fact to be highlighted is that the prediction of the churn has been an individual activity that caused difficulties when integrating or extending prediction models with other advanced intelligence techniques.

III. METHODOLOGY

In this study, the methodology is developed in four main stages, data acquisition and transformation, detection of important attributes, model construction, and model validation. Also, to evaluate the performance of the suggested model in this paper critically, the transformed and filtered data was processed with an SVM model because in previous research SVM based approaches could provide reasonable results for predicting churn [2].

A. Data Acquisition

The acquisition of data was carried out from all geographical regions of the service provider for the period of 2019-20. Following the research domain, relevant information of the customers available in ERP (which was the organizations' business operation management system) was identified and reviewed with domain experts who manage the regional and billing operations. Those attributes with details are presented in table 2. The dataset extracted includes 30,000 subscribers covering all geographical operational area of the service provider. As the suggested method is a probabilistic model, the average of the ratios between disconnected customers and active customers for the last six months was maintained while acquiring data.

Another important fact in data acquisition is the timestamp of data extraction. The suggested approach tries to predict the probability of churning before three billing cycles (each cycle usually 30 days). As illustrated in figure 3, for each subscriber the extraction point was considered as three billings cycles before the date of termination of

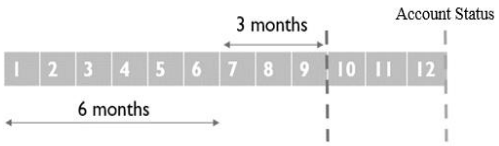


Fig. 14. Timeline of data extraction for terminated customer.

Table 2. Identified features for the prediction of churn.

Attribute	ERP Module	Description	Type
Billing Account	AR Account	Unique identification of the customer.	N/A
Connection Media	AR Account	Copper, LTE or FTTH, connection media. [COPPER, LTE, FTTH]	Categorical
Internet	AR Account	whether Subscribed for internet service or not [YES, NO]	Categorical
IPTV	AR Account	whether Subscribed for IPTV service or not [YES, NO]	Categorical
RTOM	AR Account	Geographical region of service providers' operation. (38 Regions)	Categorical
City	AR Account	City within each RTOM.	Categorical
Loyalty	AR Account	[TX date OR current date] - Subscribed date in years	Continuous
Faults	Inventory	Number of faults based on inventory issues for maintenance.	Continuous
Bill Totals	AR Invoice	Bill amounts of last 12 months	Continuous.
Payments	AR Receipts	Payment amounts of last 12 months	Continuous
Status	AR Account	[Label] Whether active (OK) or terminated customer (TX)	Target

service.

Timeline of data extraction for terminated customer.

B. Data Transformation

Even though all attributes have a potential to maintain a good relationship with the target value, bill totals and payments over a period of one year is not possible to apply directly due to following factors,

Over the time, tariff rates can be changed.

A customer has credit period to after bill generation to pay.

Usage can be high or low due to a temporary period such as Christmas or similar cultural events.

Due to the above factors, two features, Bill Totals, and payments were transformed by the following two equations. First equation (1) is the ratio of billing to payments for the last three months referred to as ratio3 while the second equation (2) is the ratio between bill to payments for the last three to six months referred to as ratio6.

$$Ratio3 = \sum_{i=1}^{3} \frac{B_i}{P_i} \quad (1)$$

$$Ratio6 = \sum_{i=4}^{9} \frac{B_i}{P_i} \quad (2)$$

C. Feature Importance

According to Jasmina Novakovic [13] in the paper titled "The Impact of Feature Selection on the Accuracy of Bayes Classifier" indicates that efficiency in all aspects of a Naïve Bayes model depends on the method of feature selection. Further, it concludes that depending on the features, the best feature selection method for the Naïve Bayes model may differ. Also, as elaborated in the related work section, there is evidence of previous research that some of the features are sensitive to market changes. Therefore, in this study minimum description method is employed before feeding the training data set into model building.

Minimum Description Length

Minimum description length (MDL) treats both models and data as codes. The key idea is that any data set can be appropriately encoded with the help of a model by uncovering underlying regularities in the data. Thus, the code length is directly related to the generalization capability of the model, where the model that provides the shortest description of the data should be chosen [14].

Following table 3, illustrates the feature importance calculated on a small sample via MDL, It indicates that according to the chosen sample attributes internet connectivity and the city has zero importance, and also it could correctly identify account number has no importance as it must be.

Table 34. Feature importance calculated via MDL algorithm

Name	Type	Rank	Importance
RATIO3	NUMBER	1	0.2971
RATIO6	NUMBER	2	0.2177
LOAYALTY	NUMBER	3	0.1731
FAULTS	NUMBER	4	0.0282
CONN_MEDIA	VARCHAR2	5	0.0099
RTOM	VARCHAR2	6	0.0063
IPTV	VARCHAR2	7	0.0005
BB	VARCHAR2	8	0.0000
BILLING_ACCOUNT	VARCHAR2	8	0.0000

Table 3 above illustrates the feature importance measured by the MDL algorithm on one instance of the data set. However, as discussed earlier in this section, the feature importance of each attribute varies based on the season. Therefore, it is possible for some attributes from the original data set to be excluded by the MDL algorithm, making refined data set with high feature importance to input for building the model.

Naïve Bayes

A Bayesian network represents the causal probabilistic relationship among a set of random variables, their conditional dependencies, and it provides a compact representation of a joint probability distribution. It consists of two major parts: a directed acyclic graph and a set of conditional probability distributions.

Mechanism of Naïve Bayes is based on joint probability distributions, for example let $\{x_1, x_2, x_3 \dots x_n\}$ be some events just like being voice only customer, already disconnected customer. In Bayesian Network, they can be represented as nodes. Now if a node has some dependency on another node then an arrow/arc is drawn from one node to another. It is interpreted as the child node's occurrence is influenced by the occurrence of its parent node. So, Bayesian Network represents a directed acyclic graph and now via conditional probability and chain rule shown in below equation (3), it is possible to get the full joint distribution i.e., the probability of the final event (churn customer) given all other dependent events.

$$P(x_1, x_2, x_3, \dots\dots, x_n) = P(x_1).P(x_2|x_1).P(x_3|x_2, x_1) \dots\dots\dots\dots P(x_n|x_1, x_{2,\dots\dots,} x_n)$$
$$= \pi(x_i|Parents(x_i)) \quad (3)$$

Zero Frequency problem

One of the disadvantages of the Naive-Bayes approach is that when there are no occurrences of a class label and a certain attribute value together in the training data set, then the frequency-based probability estimate will be zero. Hence, in a scenario where there are missing class labels, predictions can be completely wrong unless other features can overcome the zero influence.

The algorithm illustrated in figure 4 was employed to avoid the zero-frequency problem. The key idea behind the algorithm is to ensure that the categorical feature, RTOM which is a geographical attribute is fairly distributed in the training set while a numerical feature loyalty distribution is balanced in the training set.

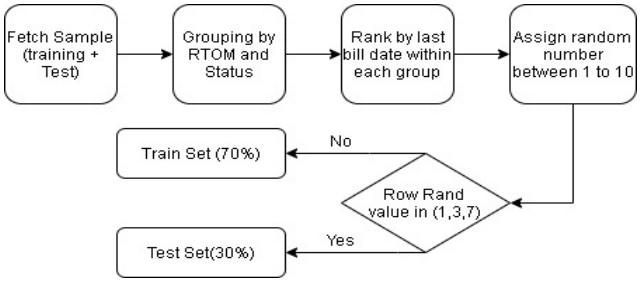


Fig.4. Flow Chart of Sampling Algorithm Used to Avoid Zero Frequency Problem

Flow Chart of Sampling Algorithm Used to Avoid Zero Frequency Problem

IV. Model Architecture

Following Fig 5 is the suggested architecture to predict customer churn in the telecommunication industry based on financial data. It consists of two major components; the first component is for automated sampling by filtering the importance of a feature at a given instance. Remaining part is the Naive Bayes model that calculates the probability of churning.

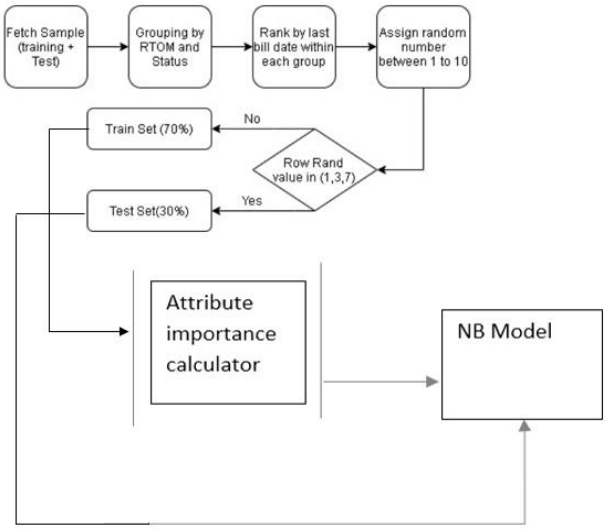


Fig. 5 Suggested Architecture of the complete prediction model

V. Results

The developed model was applied around 16,000 instances, and the performance was evaluated based on criteria related to binary classification problems which include, Rate of true positives (TP), Rate of false positives (FP), Precision, and Recall. Finally, the same set up was tested by switching the model to SVM and compare the performance between models.

A. Model Performance

Following confusion matrix summarizes the performance of the model. In this table TX refer to terminated customer, OK refers otherwise.

	OK	TX	Total
OK	6,419	1,368	7,787
TX	1,164	6,392	7,556
Total	7,583	7,760	15,343

Fig. 6 Confusion Matrix for the suggested approach

B. Confusion Matrix for the suggested approach

According to the confusion matrix illustrated in the figure 6, the performance of the suggested approach can be evaluated as follows.

Recall = TP / (TP + FN) = 84.5% (4)

Accuracy = (TP + TN) / (TP + TN + FP + FN) = 84% (5)

Precision = TP / (TP + FP) = 82.4% (6)

Specificity = TN / (TN + FP) = 82.3% (7)

Comparison with SVM

Once the NB model replaced with SVM model, efficiency was greatly reduced as the below presented confusion matrix in Fig 7.

	OK	TX	Total	Correct %
OK	5,913	3,676	9,589	61.6644
TX	4,733	4,947	9,680	51.1054
Total	10,646	8,623	19,269	
Correct %	55.5420	57.3698		

Fig. 7 Confusion Matrix with SVM model

Further, calculating classification model efficiency parameters as below highlights the drop of efficiency of prediction model due to introduction of SVM.

Recall = TP / (TP + FN) = 55.54% (8)

Accuracy = (TP + TN) / (TP + TN + FP + FN) = 57% (9)

Precision = TP / (TP + FP) = 61.2% (10)

Specificity = TN / (TN + FP) = 57.3% (11)

By comparing the results of NB model given in equations (4~7) with SVM model (8~11), the probabilistic approach of NB model has clear edge in the problem domain discussed in this paper.

C. Receiver operating characteristic curve

The receiver operating characteristic curve (ROC) summarizes the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds of each model. Following ROC clearly indicate that NB model also involve with lower trade of compared to SVM base module with respect to the problem discussed in this paper.

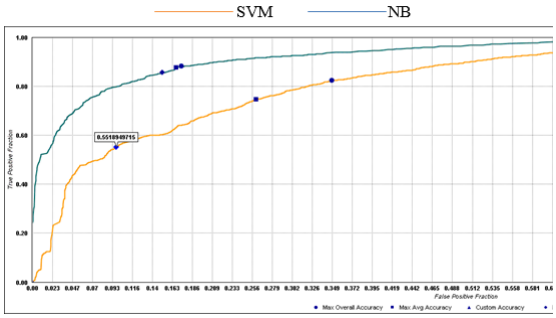


Fig. 6 ROC on Prediction [OK]

Conclusion and Future Work

In the telecommunication industry, early detection of the churn has become a crucial factor for survival. The recent development of the data science domain facilitates and provides various means to intercept the problem of the churn. However, there are no standard modules for the churn problem in any business domain as the nature of the problems and attributes vary between the environments where the business operates. It should be also noted that nowadays, many telecommunication service providers have identified that not only their revenue generation system is vital for survival, but other supportive systems are also important. Therefore, many companies tend to move towards implementing ERP systems to handle their operations. Considering all these concerns this research proposed an alternative method of predicting churn using financial information.

The study was carried out with a sample of 30, 000 subscribers, and their financial influence in the revenue generation process was analyzed to make predictions. Not like generic approaches based on consumer usage patterns that mostly govern by numerical features, this study was balanced with numerical and categorical features where there were four to numerical and six categorical features. It was the motive for selecting the probabilistic technique over logistic regression or SVM methods. The selection of the Naive Bayes method was justified at the end of the research as it yields around 85% accuracy in predicting churn.

This study also suggests that the predicting churn in telecommunication industry is not a static process, as many features related to churn is very sensitive to the change of market. Therefore, it is essential to reevaluate / fine tune models in fair frequency to keep up the accuracy of the prediction. As a solution, this paper suggests an application of minimum description length algorithm to evaluate the features of each sample before input to the model. The suggested approach is possible to apply for any dataset that has sensitivity to external variables. However, depending on the nature of the features, different algorithms may need to be employed to sustain model accuracy.

Upon completion of the research, the author concludes that similar intelligent models can be developed around the ERP system due to its integrated nature. However, it could be a challenging task to identify features to describe the problem as features could be shattered across the integrated modules. Though it could be a challenging task, building intelligence in decision making system (which is the ERP in this study) could be more effective for any industry, as it opens the way to measure decisions effectiveness and predict future impact. For example, predicting the churn in this study could be very useful for management to know the marketing waste, and compare marketing opportunity with finance status. The suggested approach is more likely building the intelligence in the same language which management system uses (such as finance data, inventory data compare to billing data like voice usage, data usage).

From the derived work so far in this research, there is a handful of future work that seems to be promising. For example, since the prediction has been done in the finance module of the ERP system, the same can be extended for revenue budgeting, forecasting, and evaluating the performance and return of marketing expenses.

REFERENCES

[1] Mbarek, R., Baeshen, Y.: Telecommunications Customer Churn and Loyalty Intention. Marketing and Management of Innovations. 110-117. 10.21272/mmi.2019.4-09 (2019).

[2] Zhao, Y., Li, B., Li, X., Liu, W., Ren, S.: Customer Churn Prediction Using Improved One-Class Support Vector Machine. 300-306. 10.1007/11527503_36 (2005).

[3] Sjarif, N.N.A., Mohd A., Nurulhuda., Sarkan, HM., Sam, S., Osman, M.: Predicting Churn: How Multilayer Perceptron Method Can Help with Customer Retention in Telecom Industry. IOP Conference Series: Materials Science and Engineering. 864. 012076. 10.1088/1757-899X/864/1/012076 (2020).

[4] Kadir, R. A., Yatin, S.: The benefits of implementing ERP system in telecommunications. Procedia-Social and Behavioral Sciences, 211, 1216-1222 (2015).

[5] Telecommunications Regulatory Commission of Sri Lanka, statistics 2019 quarter 4, http://www.trc.gov.lk/images/pdf/statis_q4_03032020.pdf, last accessed 2020/11/01.

[6] Sri Lanka Telecom PLC, Annual Report 2019. https://www.slt.lk/reports-html/annual/2019/index.html, last accessed 2020/11/01.

[7] Dahiya, K., Talwar, K.: Customer churn prediction in telecommunication industries using data mining techniques a review. Int. J. Adv. Res. Comput. Sci. Softw. Eng, 5(4), 417-433 (2015).

[8] Mozer, M., Wolniewicz, R., Grimes, D., Johnson, E., Kaushansky, H.: Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. Neural Networks, IEEE Transactions on. 11. 690 - 696. 10.1109/72.846740 (2000).

[9] Simsek G., Tugba.: Customer Churn Analysis in Telecommunication Sector. Istanbul University Journal of The School of Business Administration (2010).

[10] Saini, N., Monika., Garg, K.: Churn Prediction in Telecommunication Industry using Decision Tree. International Journal of Engineering Research and. V6. 10.17577/IJERTV6IS040379 (2017).

[11] Cui, G., Wong, M., Lui, H.: Machine Learning for Direct Marketing Response Models: Bayesian Networks with

Evolutionary Programming. Management Science. 52. 597-612. 10.1287/mnsc.1060.0514 (2006).

[12] Rouhani, S., Mehri, M.: Does ERP have benefits on the business intelligence readiness? An empirical study. 8. 81-105. 10.1504/IJISCM.2016.079559 (2016).

[13] Novakovic, J.: The Impact of Feature Selection on the Accuracy of Bayes Classifier (2010).

[14] Myung I.J.: Computational Approaches to Model Evaluation. International Encyclopedia of the Social & Behavioral Sciences (2001).

Agent Simulations in the Web Browser: A Survey of Web Browser-Based Agent Platforms

Ramesh Maddegoda[1], Asoka S. Karunananda[2]
[1]Independent Researcher,
Columbus, OH 43240, United States of America
[2] Department of Computational Mathematics, University of Moratuwa,
Moratuwa, Sri Lanka
email : [1]ramesh.maddegoda@gmail.com, [2] asokakaru@uom.lk

***Abstract*— Agent Based Modelling and Simulation (ABMS) is a very powerful technique used in many domains in order to model and simulate the actions and interactions of autonomous agents in complex systems. In the recent years, with the advancements in the frontend technologies, some of the ABMS tool developers released web browser-based ABMS tools which allow users to develop, view, share and execute agent-based simulations in any device, including the mobile devices, through a web browser. However, the existing surveys of ABMS tools mainly focus on the desktop based ABMS tools and do not provide comparisons and evaluations focused on web browser-based ABMS tools. To address this issue, we came-up with a comparison and evaluation criteria with a strong emphasis on web browser-based features. This paper presents a comparative up-to-date review of several publicly available web browser-based ABMS tools.**

Keywords*— *Agent Based Modelling and Simulation (ABMS) Tools, Swarm Intelligence, Modelling Complex Systems, Multi-Agent Systems

I. INTRODUCTION

Agent Based Modelling and Simulation (ABMS) is a promising computation modelling and simulation approach, where a system is modelled as a collection of autonomous decision making entities called agents. Each individual agent is following a basic set of rules to evaluate it's situation, making decisions and executing actions. As a result, higher level system properties emerge from the interactions between the lower level entities. The concepts of ABMS are highly inspired by the biological and sociological systems and it combines the elements of complexity science, multi-agent systems, game theory, emergence, computational sociology and evolutionary programming. ABMS is widely used in many scientific domains including biology [1], ecology [2] and social science [3].

The ABMS community has developed numerous agent based modelling tools that enable individuals to develop agent based models and simulations. Most of these tools are distributed as desktop applications and the users have to install them as software applications on computers. At the same time if an ABMS developer wants to share an agent based model with other individuals, it is necessary for those who want to view and execute the model also to have the same ABMS tool installed in their computers. As a results it is necessary to make sure that the recipients of the model are also using the same operating system, given that different ABMS tools are compatible with different operating systems. With the popularity of the smart phones, tablets and devices such as Chromebooks, there is a growing population of users who primarily use mobile operating systems and Chromebooks for their day to day activities. However, most of the ABMS tools are only supporting conventional computer operating systems and therefore it is impossible to view and execute the models built using most of the ABMS tools on mobile operating systems and Chromebooks. Also there are other reasons that discourage users from installing software on their computers such as concerns on malware and personal inconvenience.

With the advancements of the frontend web technologies, now it is possible to build web browser-based web applications that provide a look and feel, an acceptable level of performance and usability similar to the desktop applications. In recent years there were several web browser-based ABMS tools released by ABMS tool developers with the ability to execute models on web browsers. Some of those tools also provide web browser-based features to create agent based models in addition to the ability to view and execute models. As a result now it is possible for ABMS developers to share their models with others individuals, regardless of the operating systems or devices used by the recipients. Because these web browser-based ABMS tools are compatible with most of the common web browsers used in both personal computers and mobile devices.

II. RELATED SURVEYS OF AGENT BASED MODELLING AND SIMULATION TOOLS

In recent years several researchers have conducted surveys of ABMS tools. The paper "Agent Based Modelling and Simulation tools: A review of the state-of-art software" [4] presented by Sameera Abar, Georgios K. Theodoropoulos, Pierre Lemarinier and Gregory M.P. O'Hare provides a comprehensive comparative literature survey of ABMS tools and this is one of the most detailed surveys done on the subject. They have evaluated over eighty ABMS tools and provided results by comparing license/ availability, source code, type of agent based on it's interaction behaviour, programming language/API and Integrated Development Environment (IDE) used, compiler, operating system and implementation platform, modelling strength, scalability, ABMS scope or application domain. The main intention of this paper is to help scientists and engineers to quickly assess how they might choose and properly apply ABMS to their own research applications.

The "Survey of Agent Based Modelling and Simulation Tools" [5] by R.J. Allan outlines more than forty existing ABMS tools based on information found online and based on personal experiences. Also this report explores the applications of ABMS in the domains of biology and medicine, physics and chemistry, security, cyber security, the environment, social and economic modelling, supply network and transport optimisations. And this report focuses on executing ABMS on high performance computing systems.

The paper "Evaluation of free Java-libraries for social-scientific agent based simulation" [6] by Robert Tobias and Carole Hofmann compares four freely available programming libraries for support of social scientific agent based computer simulations. They have evaluated RePast, Swarm, Quicksilver and VSEit software and concluded that RePast software as a clear winner compared to the other software considered for their evaluation.

Kalliopi Kravari and Nick Bassiliades have published the article "A Survey of Agent Platforms" [7] to provide a comparative review of twenty four existing agent platforms. The authors have developed a comparison and evaluation criteria based on platform properties, usability, operating ability, pragmatics and security management. Also this article provides classifications for agent platforms in order to help readers to understand which agent platforms broadly exhibit similar properties and in which situations which choices should be made. These classifications include programming languages used by ABMS tools, FIPA compliance, application domain and the support for semantic web technologies.

"A Survey of Programming Languages and Platforms for Multi-Agent Systems" [8] by Rafael Bordini et al. presents a survey of programming languages and development tools for multi-agent systems. They have conducted this survey by considering the programming languages (declarative, imperative, and hybrid) and integrated development environments used by multi-agent systems. And finally they explore the frameworks that are not so strongly tied to a particular programming language.

Cynthia Nikolai and Gregory Madey have presented the paper "Tools of the Trade: A Survey of Various Agent Based Modelling Platforms" [9] which examines over fifty ABM platforms and characterizes each based on five important characteristics users consider when choosing a toolkit. These characteristics are: language required to program a model and to run a simulation, operating system required to run the toolkit, type of license that governs the toolkit, primary domain for which the toolkit is intended, and types of support available to the user.

Many of the above surveys provide very comprehensive and detailed comparisons of ABMS tools. Some of the above surveys have considered few web browser-based ABMS tools also in their research. However, the main focus of above surveys were not to evaluate the features specific to web browser-based ABMS tools. Our intention was to conduct a survey of web browser-based ABMS tools in order to fill this gap and provide a supplement to the existing surveys of ABMS tools.

III. Overview of the Web Browser-Based Agent Platforms

At the time of this survey, following web browser-based ABMS tools were publicly available on the internet and considered for the survey.

A. AgentBase

AgentBase.org [10], [11] allows users to build agent based models that run in the web browser. It follows the Agent oriented Programming model of NetLogo [12]. Models in AgentBase use the AgentBase Library, and are written in CoffeeScript, which is instantly interpreted as JavaScript. This AgentBase library was derived from AgentScript and it provides a rich set of resources for moving and drawing agents, neighbour detection, and many other features.

B. AgentScript

AgentScript.org [13] provides a minimalist agent based modelling system based on NetLogo semantics. It has a Model View Control (MVC) architecture and it is a modern ES6 module based JavaScript implementation.

C. Behaviour Composer

Behaviour Composer [14], [15] is a web based tool available on modelling4all.org to support teachers, learners and researchers, including those with little or no programming experience, to build, share, and discuss agent based computer models using a visual tool. When a user executes a model built using the Behaviour Composer, it automatically converts the model in to a NetLogo file and loads the model in to a web browser-based view powered by NetLogo Web [16].

D. Insight Maker

Insight Maker [17], [18] is a modelling and simulation tool that runs in web browser. It provides a rich set of features for system dynamics modelling and agent based modelling. Also insightmaker.com provides a rich collection of models called Insights, published by Insight developers.

E. Mesa

Mesa [19], [20] is an agent based modelling framework for Python. Mesa allows users to build agent based models, visualize them using a web browser-based interface and analyse the results using Python related data analysis tools. With the recent popularity of Python based tools, Mesa provides a low learning curve for the users who are already familiar with Python.

F. NetLogo Web

NetLogo Web [16] is a version of the NetLogo modelling environment [12] that runs entirely in the web browser. NetLgo is one of the most widely used and feature rich ABMS tool. At the moment NetLogo Web provides only a subset of the features available in NetLogo desktop version. However, NetLogo Web is a highly usable and stable ABMS environment which provides most of the necessary features.

G. StarLogo Nova

StarLogo Nova [21] is the online version of StarLogo TNG [22], which is a widely known ABMS language. It provides a web browser-based environment to build and

simulate models. StarLogo Nova mainly focuses on education with an effort to introduce programming skills to younger students. It uses a programming language of coloured blocks that fit together like puzzle pieces. StarLogo Nova is used by users with various backgrounds to create games, simulations and to study diverse concepts in science and maths, such as epidemiology, ecology, geometry and computational thinking.

IV. Comparison and Evaluation Criteria

Based on our literature review on previous surveys of ABMS tools, we identified that the comparison and evaluation criteria of previous surveys have not addressed some of the important accepts specific to web browser-based ABMS tools. Therefore we came-up with our comparison and evaluation criteria based on 1) Constraints, web browser compatibility and supported devices; 2) Licence, cost, current development status and affiliations; 3) Programming languages and terminology; 4) Access control and model sharing; 5) Availability of classic agent based models, and 6) Learning resources, user community and model repositories. The following sections of this paper review the selected ABMS tools with the help of this comparison and evaluation criteria.

V. Constraints, Web Browser Compatibility and Supported Devices

One of the main advantages of web browser-based ABMS tools is the ability to share and execute agent based simulations on web browsers regardless of the operating system or the device of the user. In addition to the ability to execute agent based simulations, some of the ABMS tools included in this survey provide features to create and edit agent based models only using a web browser. For example, AgentBase, Behaviour Composer, Insight Maker, NetLogo Web and StarLogo Nova allow users to create and edit models on web browser without any additional tools. On the other hand, AgentScript and Mesa are primarily software libraries to be used with programming languages such as JavaScript and Python and they do not provide in-built features to edit the models only using a web browser. When using AgentScript or Mesa, it is expected a developer to create and host the agent based simulations as a web application, so other individuals can view and execute agent based simulations through web browsers. However, since Mesa is a Python package, it is possible to import Mesa through a Jupyter Notebook [23]. A Jupyter Notebook is an open source web application that enables users to create and share documents that contain live code, equations, visualizations and text. When Mesa is imported in a Jupyter Notebook, it provides an interactive environment where users can create, edit and execute agent based models and simulations in a web browser.

Compared to the desktop versions of ABMS tools, the web browser-based ABMS tools provide a limited set of features due to the constraints exist with web browsers. For example, NetLogo desktop version provides a 3D (three-dimensional) view where agent based modelling and simulation can be executed in a 3D environment. However, the NetLogo Web version only supported a 2D (two-dimensional) view at the time of writing this paper. Another example is the GIS (Geographic Information Systems) support available as an extension to the NetLogo desktop version. At the time of this survey, NetLogo Web did not have a support for extensions such as GIS extension. Table 1 provides a comparison of ABMS tools in terms of 2D, 3D and GIS support.

Table 1 2D, 3D and GIS Support

ABMS Tool	View Agent Based Simulations in 2D	View Agent Based Simulations in 3D	GIS Support
AgentBase	✓	-	-
AgentScript	✓	✓	✓
Behaviour Composer	✓	-	-
Insight Maker	✓	-	-
Mesa	✓	-	✓ (with mesa-geo GIS extension)
NetLogo Web	✓	-	-
StarLogo Nova	✓	✓	-

To test the compatibility with different web browsers, it was decided to evaluate each ABMS tool with four widely used web browsers available on desktop computers and laptops. The ability to create models, ability to edit models and ability execute simulations of each tool were tested with different web browsers. For AgentScript, only the ability to execute simulations was tested, because AgentScript is a software library that does not provide any interactive editing tools to create or edit models using a web browser. The Mesa ABMS tool, which is another software library, was tested after importing Mesa in an interactive Jupyter Notebook. All the ABMS tools considered for this survey passed this web browser compatibility test with Google Chrome, Microsoft Edge, Mozilla Firefox and Safari web browsers.

With the growing use of the mobile phones, tablets and devices such as Chromebooks, there is a very high probability that a given user wants to view an agent based simulation on a mobile device or a Chromebook. Therefore it was decided to test all of these ABMS tools by viewing similar models on different types of mobile devices and Chromebooks. Therefore multiple Android phones, Android tablets, iPhones, iPads and Chromebooks were used for this test. All the ABMS tools considered for this survey passed this mobile browser and device compatibility test with Google Chrome and Mozilla Firefox web browsers on both Android and iOS (iPhone and iPad) devices, Safari web browser on iOS (iPhone and iPad) devices and Google Chrome web browser on Chromebook devices.

VI. Licence, Cost, Current Development Status and Affiliations

The type of the software licence of each ABMS tool, the cost, current development status and affiliations are very important factors when selecting an ABMS tool. These factors are very critical to the long-term success of an ABMS project which utilizes an ABMS tool. All the ABMS tools considered for this survey were available free of charge, at the time of this survey. However, there are differences in the software licences used and these differences in the licenses are important for developers who wish to distribute derived work from the source code of these ABMS tools. Table 2 presents a comparison of licences, current development status related information and affiliations for each ABMS tool.

Table 2 Licence, Current Development Status and Affiliations

ABMS Tool	Licence	Current Development Status and Affiliations
Agent Base	GNU General Public License, version 3	Latest Update: Feb, 2017 Source Code: https://github.com/wybo/agentbase Copyright (c) Wybo Wiersma, 2014 Affiliation: University of Oxford
Agent Script	GNU General Public License, version 3	Latest Update: Oct, 2020 Source Code: https://github.com/backspaces/agentscript Copyright (c) Owen Densmore, RedfishGroup LLC, 2012-2020 Affiliation: RedfishGroup LLC
Behaviour Composer	BSD License	Latest Update: May, 2015 Source Code: https://code.google.com/archive/p/modelling4all/source Written by Dr. Ken Kahn, Copyright (c) University of Oxford, 2014 Affiliation: University of Oxford
Insight Maker	Insight Maker Public License (a custom licence based on the Affero GPL)	Latest Update: October, 2020 Source Code: https://github.com/scottfr/insightmaker Copyright (c) Scott Fortmann-Roe, 2010-2020 Affiliation: insightmaker.com
Mesa	Apache License, Version 2.0	Latest Update: October, 2020 Source Code: https://github.com/projectmesa/mesa Copyright (c) Core Mesa Team, 2020 Affiliation: Project Mesa
NetLogo Web	GNU General Public License, version 2	Latest Update: September, 2020 Source Code: https://github.com/NetLogo/NetLogo Copyright (c) Uri Wilensky, 1999-2014 Affiliation: Northwestern University Center for Connected Learning and Computer-Based Modelling
StarLogo Nova	License information is not publicly available	Latest Update: November, 2018 StarLogo Nova source code and copyright details were not publicly available at the time of writing this paper Affiliation: Massachusetts Institute of Technology (MIT) Scheller Teacher Education Program

VII. Programming Languages and Terminology

The programming language used in an ABMS system in order to develop agent models is also a key deciding factor when selecting an ABMS tool. In some of the domains where agent based modelling and simulations are highly useful, such as biology and social science, the domain experts are not very familiar with the traditional programming languages used for computer programming. In such cases, visual programming approaches provided by some of the ABMS tools are preferred and some of these visual programming approaches are very convenient for users to successfully develop agent based models and simulations, with limited or no programming experience. Table 3 provides a comparison of the different programming languages used in the selected ABMS tools.

Table 3 Programming Languages Used in Selected ABMS Tools

ABMS Tool	Programming Language	Level of Traditional Programming Skills Required
AgentBase	CoffeeScript	High
AgentScript	JavaScript	High

Behaviour Composer	Provides a visual composer with visual elements such as prototypes, micro behaviours and attributes Also it is possible to add NetLogo code through micro-behaviours	Low
Insight Maker	Enables users to build models graphically and supports a rich set of diagramming features.	Low
Mesa	Python	High
NetLogo Web	NetLogo	Medium
StarLogo Nova	A blocks-based programming language, which uses blocks to put together puzzle-like pieces	Low

It was observed that many of the ABMS tools included in this survey were inspired by NetLogo and provides NetLogo like semantics for agents. Table 4 shows the terminology used by different ABMS tool to represent agents.

Table 4 Agent Based Modelling Terminology Used in Different ABMS Tools

ABMS Tool	Agent Based Modelling Related Terminology
AgentBase	Provides NetLogo like semantics for agents: *Turtles*, *Patches* and *Links*
AgentScript	Provides NetLogo like semantics for agents: *Turtles*, *Patches* and *Link*
Behaviour Composer	Agent are represented as *Prototypes*. *Attributes* and *Micro-behaviours* can be added to the *Prototypes*
Insight Maker	Agent Based Models are primarily built using *States*, *Transitions*, *Actions*, *Variables* and *Agent Populations*
Mesa	Agents are represented with the *Agent* class
NetLogo Web	Four types of agents are available: *Turtles* are agents that move around in the world, *Patches* have locations with coordinates, *Links* are connecting turtles and *Observers* give instructions to other agents
StarLogo Nova	Agents are represented as *Turtles*

VIII. Access Control and Model Sharing

Some of the ABMS tools require users to create a user account, in order to use the ABMS tool. These user account are used for access control and provide a persistent storage in some of the tools. Also this determines the ways a user can share models with other users. Table 5 compares the access control, storage and sharing related features of ABMS tools.

Table 5. Access Control, Storage and Sharing Related Features of ABMS Tools

ABMS Tool	User Account Required?	Can Store Models in an Online Storage?	Ways to Share Models
AgentBase	Yes	Yes	As a URL
AgentScript	No	No	As a URL
Behaviour Composer	No	No	As a URL to be automatically opened with NetLogo Web, embedded as iFrame or embedded as a link in a web site
Insight Maker	Yes	Yes	Embed Insight (model) in a website, export as a file or share as a *Storytelling* (step-by-step walk-through of the model.)
Mesa (with Jupyter notebook)	Depends on the access control of the Jupyter notebook	Depends on the access control of the Jupyter notebook	As a Jupyter notebook or as a URL
NetLogo Web	No	No	As a URL or export as a NetLogo file
StarLogo Nova	Yes	Yes	As a URL

IX. Availability of Classic Agent Based Models

All of the ABMS tools considered for this survey were available with a collection of sample agent based models. In addition to that some of the popular ABMS tools have community based online model repositories with agent based models created by community members. The availability of classic agent based models built using a given ABMS tool increases confidence of the users on the abilities of the ABMS tool. Table 6 shows the classic agent based models used to compare ABMS tools on the availability of classic agent based models.

Table 6 . List of Classic Agent Based Models Used for the Survey

Model	Description
Boids	*Boids* is an artificial life program, developed by Craig Reynolds [24] in 1986, which simulates the flocking behaviour of birds

Life	*Game of Life* is a cellular automaton devised by John Horton Conway [25] in 1970. It is an example of a zero-player game where the evolution is determined by its initial state, requiring no further input
Fire	*Forest Fire* model [26] simulates the spread of a fire through a forest. This model is defined as a cellular automaton on a grid with cells. A cell can have one of the three states: empty, occupied by a tree, or burning
Epidemic	*Epidemic* model [27] is a simplified simulation of the spread of an infectious disease in a closed population
Ants	*Ants* model [28] simulates a colony of ants forage for food. This model is an interaction between food sources, ants and chemicals dropped by ants
Segregation	*Schelling's Model of Segregation* is an agent based model developed by economist Thomas Schelling [29]
Predator and Prey	*Predator and Prey* model explores the stability of a predator and prey ecosystem. Wolf Sheep Predation [30] is a specific version this model which involves wolves and sheep
Prisoner's Dilemma	*The Prisoner's Dilemma* [31] is a popular example of a game analysed in game theory that shows why two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so

Table 7 compares the public availability of above mentioned classic agent based models built using the ABMS tools at the time of conducing this survey.

Table 7 Availability of Selected Classic Agent Based Models Built Using the ABMS Tools

ABMS Tool	Boids	Life	Fire	Epidemic	Ants	Segregation	Predator and Prey	Prisoner's Dilemma
AgentBase	✓	✓	-	-	✓	-	-	-
AgentScript	✓	-	✓	-	✓	-	-	-
Behaviour Composer	-	-	-	✓	-	-	✓	-
Insight Maker	-	✓	✓	✓	-	✓	✓	✓
Mesa	✓	✓	✓	-	-	✓	✓	✓
NetLogo Web	✓	✓	✓	✓	✓	✓	✓	✓
StarLogo Nova	✓	-	✓	✓	✓	✓	✓	✓

X. Learning Resources, User Community and Model Repositories

Availability of rich learning resources and user community are some major considerations when selecting an ABMS tool. Also the availability of model repositories helps users to view and learn from the models developed by other individuals. Table 8 presents the availability of learning resources, user community and model repositories.

Table 8 Learning Resources, User Community and Model Repositories

ABMS Tool	Amount of Tutorials and Documentation	Dedicated User Community Forum	Model Repositories
AgentBase	Small	-	Only a small number of models distributed with the tool
AgentScript	Small	-	Only a small number of models distributed with the tool
Behaviour Composer	Medium	✓	Several models and micro behaviours are available with Modelling4All Project
Insight Maker	Medium	✓	A large number of models created by users are available through the Explore Insight page of the insightmaker.com
Mesa	Small	-	Only a small number of models distributed with the tool
NetLogo Web	Large	✓	Over 1000 models created by users are available through multiple repositories including NetLogo User Community Models [32] and Modeling Commons [33]
StarLogo Nova	Large	✓	A large number of models created by users are available through the Star Logo Nova projects search

XI. Conclusion

Based on the findings of this survey it appears that each ABMS tool has both strengths and weaknesses based on the requirements of the user. All of the ABMS tools provide the ability to view and execute agent based models and simulations on web browsers. And all of these tools (except AgentScript) allow users to create and edit models on web browser. And every single ABMS tool selected for this survey supports 2D views on web browsers. However, only AgentScipt and StarLogo Nova provide 3D views on web browsers. If an agent based model requires support for GIS, it is advisable to use AgentScipt or Mesa (with mesa-geo GIS extension) tools which support GIS. All selected ABMS tools were available free of charge at the time of this survey and as a result there were no costs associated with using these tools.

The programming languages used to develop agent based models and simulations with these tools were compared and presented in this paper. This comparison helps users to select ABMS tools with the preferred

programming language. Also there are some ABMS tools available with visual programming environments to help users with little or no programming experience to effectively build agent based models and simulations. On the other hand, experienced developers may prefer to use a JavaScript, CoffeeScript or Python based ABMS tools with the flexibility to integrate those tools with existing software stacks. Furthermore, we reviewed the access controls available in each tool and ways to share models with others.

Finally, we compared the availability of classic agent based models, learning resources, user community forums and model repositories for each ABMS tool that can heavily influence a decision to select an ABMS tool for a long-term project. We believe that the comparisons provided by this survey will provide guidelines for individuals planing to use web browser-based ABMS tools. Also we hope this survey will provide a supplement to the existing surveys of ABMS tools.

REFERENCES

[1] Bora, S., Emek, S.: Agent-Based Modeling and Simulation of Biological Systems. In: Modeling and Computer Simulation, Dragan Cvetković, IntechOpen (2018)

[2] Grimm V., Railsback S.F.: Agent-Based Models in Ecology: Patterns and Alternative Theories of Adaptive Behaviour. In: Billari F.C., Fent T., Prskawetz A., Scheffran J. (eds) Agent-Based Computational Modelling. Contributions to Economics, pp 139--152. Physica-Verlag HD (2006)

[3] Klein, D., Marx, J., Fischbach, K.: Agent-Based Modeling in Social Science, History, and Philosophy. An Introduction. In: Historical Social Research 43 (2018)

[4] Abar, S.,Theodoropoulos, G., Lemarinier, P., O'Hare, G.:Agent Based Modelling and Simulation tools: A review of the state-of-art software. In: Computer Science Review (2017)

[5] Allan, R.J.: Survey of Agent Based Modelling and Simulation Tools. In: Technical Report DL-TR-2010-007, Science and Technology Facilities Council (2009)

[6] Tobias, R., Hofmann, C.: Evaluation of free Java-libraries for social-scientific agent based simulation. In: Journal of Artificial Societies and Social Simulation, vol. 7 (2004)

[7] Kravari, K., Bassiliades, N.: A Survey of Agent Platforms. In: Journal of Artificial Societies and Social Simulation. 18 (2015)

[8] Bordini, R., Lars, B., Dastani, M., Seghrouchni, A., Gómez-Sanz, J., Leite, J., O'Hare, G., Alexander, P., Ricci, A.: A Survey of Programming Languages and Platforms for Multi-Agent Systems. In: Informatica 30 (2006)

[9] Nikolai, C., Madey, G.: Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. In: Journal of Artificial Societies and Social Simulation, vol. 12 (2009)

[10] Wiersma W.: AgentBase: Agent Based Modeling in the Browser. In: Jager W., Verbrugge R., Flache A., de Roo G., Hoogduin L., Hemelrijk C. (eds) Advances in Social Simulation 2015. Advances in Intelligent Systems and Computing, vol 528. Springer, Cham (2017)

[11] AgentBase, http://agentbase.org/

[12] Tisue, S., Wilensky, U.: NetLogo: A simple environment for modeling complexity. In: International Conference on Complex Systems (2004)

[13] AgentScript, http://agentscript.org/

[14] Kahn, K. Noble, H.: The Modelling4All project a web-based modelling tool embedded in Web 2.0. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, SimuTools (2009)

[15] Modelling4All Project, http://m.modelling4all.org/

[16] NetLogo Web, https://www.netlogoweb.org/

[17] Fortmann-Roe, S.: Insight Maker: A general-purpose tool for web-based modeling & simulation. In: Simulation Modelling Practice and Theory, vol. 47, pp. 28--45 (2014)

[18] Insight Maker, https://insightmaker.com/

[19] Masad, D., Kazil, J.: Mesa: An Agent-Based Modeling Framework. In: Python in Science Conference (2015)

[20] Mesa: Agent-based modeling in Python 3+, https://mesa.readthedocs.io/en/master/

[21] StarLogo Nova, https://www.slnova.org/

[22] Klopfer E., Scheintaub H., Huang W., Wendel D.: StarLogo TNG: Making Agent-Based Modeling Accessible and Appealing to Novices. In: Komosinski M., Adamatzky A. (eds) Artificial Life Models in Software. Springer, London (2009)

[23] Project Jupyter, https://jupyter.org/

[24] Reynolds, C. W.: Flocks, Herds, and Schools: A Distributed Behavioral Model. In: Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings), pp. 25--34. (1987)

[25] Gardner, M.: Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life". In: Scientific American, vol. 223, pp. 120--123 (1970)

[26] Wilensky, U.: NetLogo Fire model, Center for Connected Learning and Computer-Based Modeling, Northwestern University, http://ccl.northwestern.edu/netlogo/ models/Fire

[27] Yang, C., Wilensky, U.: NetLogo epiDEM Basic model, Center for Connected Learning and Computer-Based Modeling, Northwestern University, http://ccl. northwestern.edu/netlogo/ models/epiDEMBasic

[28] Wilensky, U.: NetLogo Ants model, Center for Connected Learning and Computer-Based Modeling, Northwestern University, http://ccl.northwestern.edu/netlogo/ models/Ants

[29] Schelling, T.C.: Dynamic models of segregation. In: The Journal of Mathematical Sociology, vol. 1, pp. 143--186 (1971)

[30] Wilensky, U.: NetLogo Wolf Sheep Predation model, Center for Connected Learning and Computer-Based Modeling, Northwestern University, http://ccl.northwestern. edu/netlogo/ models/ WolfSheepPredation

[31] Prisoner's Dilemma, https://plato.stanford.edu/entries/prisoner-dilemma/

[32] NetLogo User Community Models, http://ccl.northwestern.edu/netlogo/models/ community/index.cgi

[33] Modeling Commons, http://modelingcommons.org

Artificial Neural Network Based Bionic Leg for Upper Knee Leg Amputation

A P T D Pathirana,
Department of Mechanical Engineering, Faculty of Engineering Technology,
The Open Universirty of Sri Lanka, Nugegoda, Sri Lanka
email: thisarapathirana@ieee.org

***Abstract*— Amputation of the leg is a common factor in military accident cases. Mainly leg amputations can be divided into two broad categories - minor amputations and major amputations. Minor amputations generally refer to the amputation of digits. Major amputations are commonly referred to as below-knee amputation and upper-knee amputation.**

Most of the commercial bionic legs are act on the electromyography (EMG) signals that detect from the thigh muscles. If the amputation happens in the upper knee by causing damage on thigh muscles such as military accidents made difficult to detect EMG signals. So, human who faced leg amputation with having damaged or inactive of thigh muscles, current commercial bionic legs are unable to use. Therefore, this research produces the active bionic leg which was controlled by the movement of human arm using Artificial neural network (ANN) for leg amputees with having inactive muscles of thigh.

The Model based technique such as artificial neural network (ANN) will be a pragmatic solution for this kind of randomness inherited cases. Generally, ANN is acting as the function to make relationship between input and corresponding output. This paper discusses a novel approach for developing a bionic leg which controlled by upper limb motion using artificial neural network model. This research focusses on analyzing the arm swing pattern related to the human gait cycle and develop an ANN model using upper limb motion (arm swing motion) related to the lower limb motion (leg motion). ANN model for whole system was validated using a prototype. According to the citation for results of healthy person, average gait cycle time was 1070 milliseconds. Bionic leg has a capability to be predict the corresponding knee joint angle using human arm swinging pattern around the 13 milliseconds with reference to the general human gait cycle time of 1070 milliseconds. This execution time is added to the general gate cycle time to complete full gate cycle for disable person. As a result of, gait cycle with bionic leg may have a time delay deviation by 1.21% between human gait cycle of normal healthy person and a disable person who wearing the bionic leg

Keywords—Active bionic leg, Artificial neural network, Human gait cycle time

I. INTRODUCTION

Below knee leg amputation can be solve using mechanical prosthetic leg without any sophisticated controlling parts. In point of view of major knee amputation, knee motion rehabilitation was the challenged in active prosthetic legs with having advanced controlling parts. Mechanical passive prosthetic legs are supported to the static stability for the disabled persons.

According to the literature review, most kind of passive prosthetic legs were developed for the below knee (BK, transtibial) mode [2]. But its unable to use against body dynamic balance during the bipedal walking [2, 3, 16, 22]. Active bionic legs come to solve this problem with more cutting edge technologies [19, 20, 21]. When discuss about concepts on current commercial active bionics legs, more sensitivity was included for the control signals using advanced controlling method including the locomotion of knee joint in bionic leg [4, 13].

Human who faced the leg amputation with having damaged of inactive muscles of the thigh, current commercial bionic leg was unable to use due to difficulty of gathering of the EMG signal on the thigh. This research discusses the design and implementation of a bionic leg based on the pre-trained model using Artificial Neural Networks (ANN) to predict the knee motion according to the arm swinging signal.

ANN is used as a random function approximation tool. This help to estimate the most effective and ideal methods for arriving at solutions while defining computing functions or distributions [5]. Training an artificial neural network involves choosing from allowed models for which there are several associated algorithms. The movement of arm are maintained by the muscles group in the human body and can be recognized using electromyography (EMG). This arm movement also can be tracked using the motor signal which generated in human brain and central pattern generator. But rhythmic pattern of the human body, like arm swinging was unable to detect properly using EEG. In this research project, Motion of the arm can be obtained as direct measurements using (Inertial Measurement Unit) IMU sensor. IMU sensor can be mount on surface of human arm and it never communicate with muscle to obtain the angle data. movement of the human limbs can be tracked using special purpose sensor and muscle activity based active limbs are best solution for the amputation for the human limbs. Active bionic limbs use these methods to get corresponding input from human body.

Vastus muscles and Rectus femoris muscles in human thigh [1, 6, 7] were damaged against the upper knee leg amputation with huge damaged of the thigh. In this case, Existing bionic leg products were difficult to use because of acquisition of the EMG signals from the inactive or damaged muscles is more difficult. Furthermore, exact signal pattern was unable to find for the particular task in walking gait cycle by the sensor on inactive muscles. Also more signal processing concepts and pattern recognition algorithms make slow response for the bionic leg [8].

Every human swing their arms to make the gait stability during the bipedal walking. The Central pattern generator of the human body concerned that gait stability by making the motion of arm corresponding walking patterns. [9, 10, 11, 16]. This bionic leg has capability to predict knee motion according to the arm movement during the bipedal walking of human.

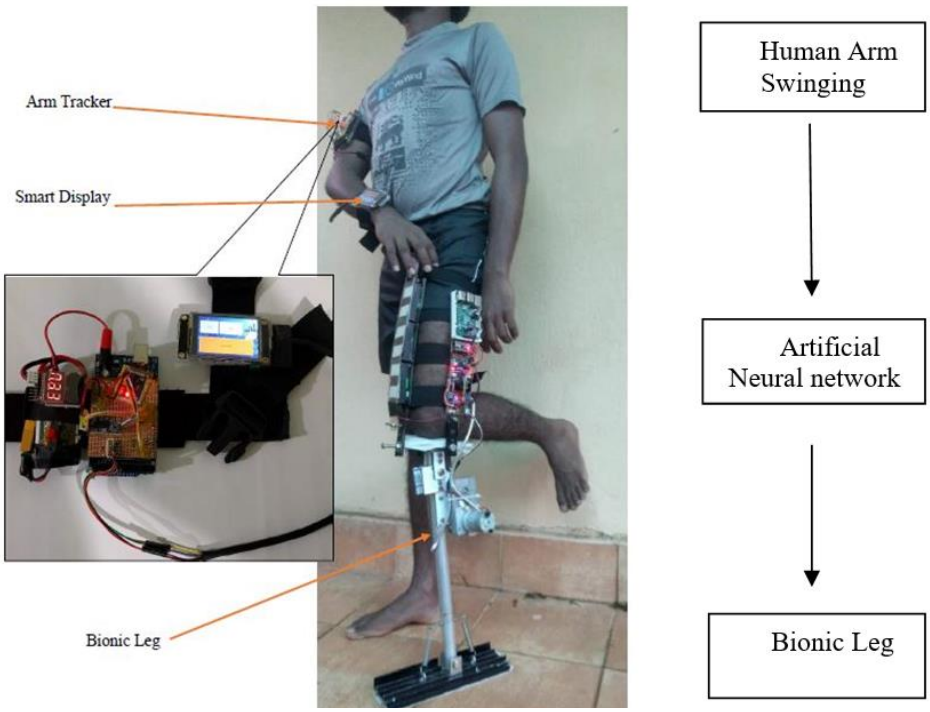


Fig. 1 ANN based bionic leg model

Figure 1 shows the major parts of the bionic leg with the (ANN) which was developed as the execution software. Arm swing pattern was grasped using arm tracker shown in Figure 1. The obtained arm swing pattern fed in to the pre-trained

Artificial Neural Network model and ANN has capability to predict the corresponding knee angle for the bionic leg. The predicted knee angle by ANN was the desired input for the knee motion controller of the bionic leg and knee motion controller make the knee position similar to the desired input (Predicted knee angle by ANN according to the arm swing patterns).

II. METHODOLOGY

The development of the bionic leg is carried out in two stages. Also, this ANN based bionic leg can predict the knee motion in related to the arm motion of human during bipedal walking. The main challenge of this project was the eliminate the knee motion for the unwanted arm movement. This problem was solved by using the ANN by considering the rate of change of arm angle during the bipedal walking. The multilayer ANN was trained under the backpropagation method. Initially Artificial Neural Network was developed and then fabricated the bionic leg according to the results of ANN. Human arm swinging data was tracked by the sensors and corresponding lower limb data such as thigh angle and knee angle were tracked in same time during the bipedal walking in natural speed by using wearable sensors as Figure 1. Second stage was validating present concept using the development of bionic leg Prototype using mechanical engineering, control engineering theories and practices [14, 15, 17, 18]. The model will be developed only for the walking on the flat surface. For this study Artificial Neural Network model was developed for healthy persons.

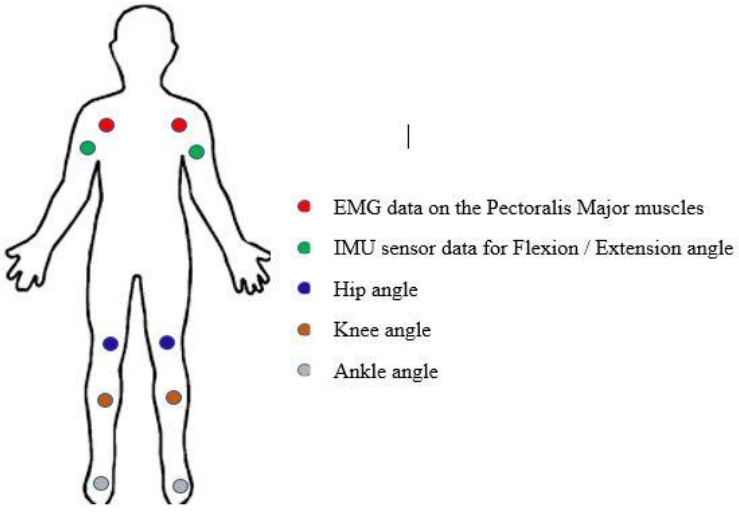


Fig. 2. Wearable sensor attachment on human body for the Data acquisition

A. Data acquisition

Data acquisition system contribute to make training data using EMG Sensors and accelerometer IMU Sensor by contacting the specific areas of the human body to make the data for train Artificial Neural Network. Figure 2 represent the attachment of the wearable sensors which combination of the IMU sensors and EMG sensors. data acquisition system

was programmed by using microcontroller and sensorized data should be processed using computer with data processing algorithm. EMG sensor and IMU sensor were attached to the microcontrollers because of the response of the data acquisition should be expedite during the bipedal walking. Microcontroller and microcontroller's boot loading platform are based on the Arduino Mega that powered by ATmega2560 AVR (AVR is a family of microcontrollers developed by Atmel beginning in 1996) Microcontroller with 16 MIPS CPU speed at 16 MHz and operates between 4.5-5.5 volts. Data acquisition part was executed on selected microcontroller with combination of the EMG sensor and number of 8 IMU sensors that comes from the one wire data bus.

adjusted for each application using the weight factor with having Exponential term.

III. Artificial Neural Network model

A. ANN Architecture

Four inputs and one output were existed on the neural network to operate the bionic leg's knee. Inputs of the ANN In01, In02, In03 and In04 as shown in Figure 4 are represented the body weights, body height, moving rate of change of Arm and IMU Angle value on the Arm respectively during the walking. Generally, Body weight and Body height are the constant value for the specific human also moving rate values and Angle values on Arm were changed with time during the walking with arm swinging.

Providing the above mentioned inputs during the walking with arm swinging, Trained Artificial Neural Network was capable to predict the output of knee angle of the bionic leg. These output were feed to the PID controller of bionic leg to operate the leg movement using arm swinging pattern using the above Artificial Neural Network architecture shown in Figure 4.

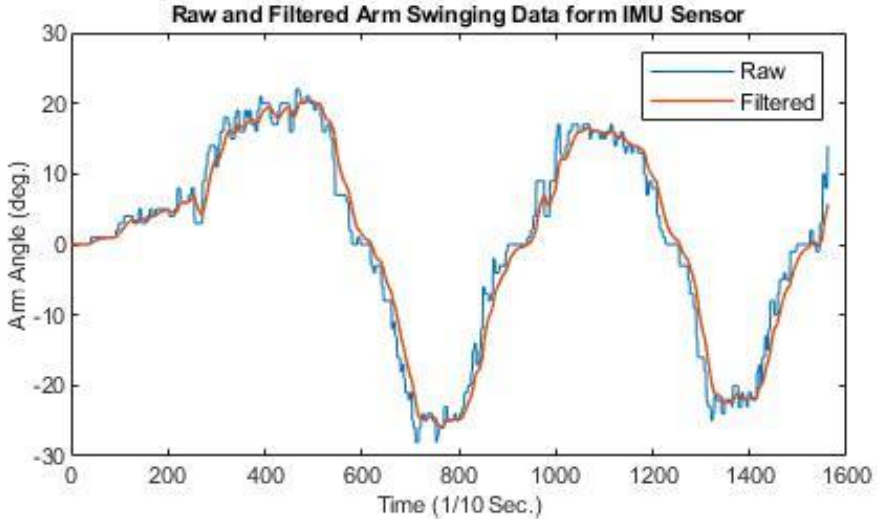


Fig. 3 Raw data and filtered data - Arm swinging

During the acquisition part of the signal form IMU sensors and EMG sensor, the smoothness of the signal was important factor to prepare the input output data for the neural network development.

The Exponential filter is a recursive filter. A recursive filter is just one that calculates a new, smoothed value by using the last smoothed value and a new measurement [12]. Two successive data sample is used to filter the data in the exponential filter algorithm.

Filtered data (Figure 3) can be gathered simultaneously during the real time data acquisition process. The small execution delay for the filtering process between raw data and filtered data was existed by using the Exponential filter algorithm. Delay time of the filtering process was eliminated by using exponential filter and filtering smoothness can be

Proposed Artificial Neural Network which was summarized in the Figure 4 was developed using the "Python" Programming language under the "Google TensorFlow™" Machine learning tool library. Supervised learning method was used to train the Neural Network. Multilayer flat (2D) ANN was trained using backpropagation method.

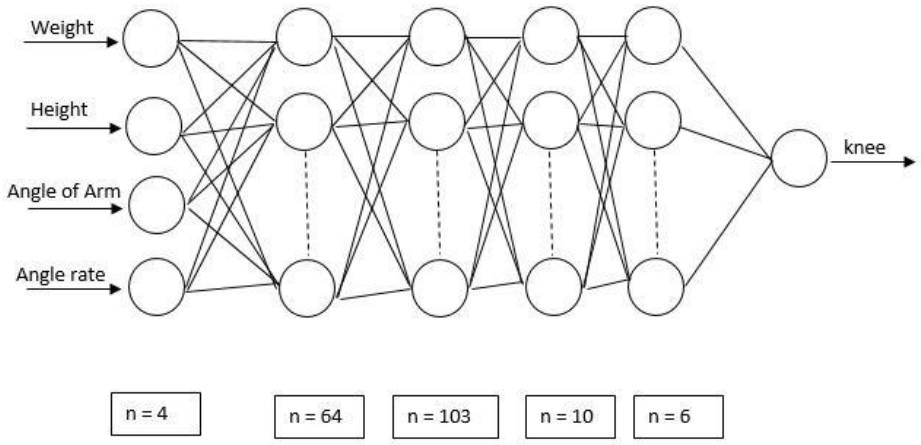


Fig. 4 ANN Architecture with 4 inputs and 1 outputs

B. Accuracy and loss optimization

After the developed an Artificial Neural Network by using the random architecture with having random Hyper-parameters (Weights, Bias, Layers etc.), architecture was optimized to obtain the good accuracy variation and less number of loss variation.

Fluctuated accuracy variation and loss function variation was represented in first attempt of training. After the adjustment of the number of Hidden Layers and Neurons on each layers, accuracy and loss was smoothed without any fluctuation. Currently, Artificial Neural Network has around 66% accuracy and around 26% loss value for the Number of three Hidden layers and with having architecture earlier. Also accuracy and loss values can be improved after the more adjustment of the Hyper-parameters of the Artificial Neural Network. Furthermore, Final accuracy of the ANN was obtained 97% for the architecture which was represented in Figure 4.

C. ANN Validation

The K-Fold Cross-Validation method was the kind of cross validation which was used in this research. The data set is separated into two sets, called the training set and the testing set. The function approximation tool fits a function using the training set only. Then the function approximation tool is asked to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made. In this case, data was split into 70% training and 30% Testing along four number of training event for the k-Fold cross validation. (K = 4 training event).

IV. RESULTS

Terminal Stance and Pre-Swing phase in human gait cycle were represented with the movement of bionic leg during the gait cycle in Figure 5(a) and Figure 5(b) respectively.

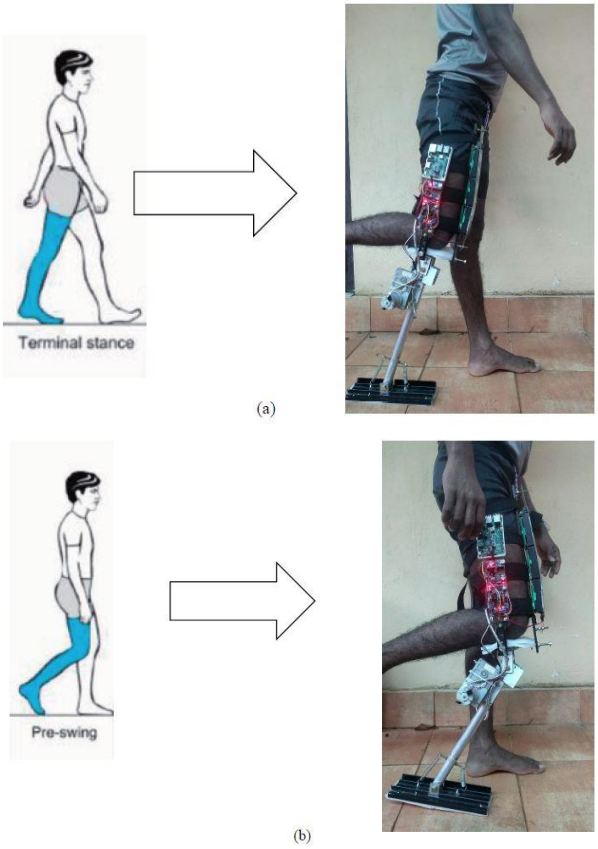


Fig. 5 Human walking pattern in special phases in Human gait cycle

Average gait cycle time of healthy human (young male) were 980 to 1070 milliseconds [23]. According to this literature, Average gait cycle time of 1070 milliseconds can be considered as the reference in this research for the analysis.

Gait cycle time (BL) = Gait cycle time (Human) + Soft. Execution delay + Communication delay + PID **(1)** *Control delay*

By considering the numerical data which were obtained in this research on Equation 1, Gait cycle time (Human), Software execution delay with communication delay and PID control execution delay were 1070, 5 and 8.17 milliseconds respectively. It seems additional 13.17 milliseconds ware added to average gait cycle time of human. Then gait cycle time of the bionic leg was around 1083 milliseconds.

Terminal Stance and Pre-Swing phase in human gait cycle were represented with the movement of bionic leg during the gait cycle in Figure 6. Arm data curve was displayed with corresponding actual knee motion (Actual. Knee) during the gait cycle and predicted data form the Artificial Neural Network (ANN) (ANN.knee), furthermore knee motion form the Knee motion PID controller in the Bionic Leg (BioLeg.Knee). According to the graph in Figure 6, Root Mean Square Error (RMSE) between actual knee (Actual.Knee) angle and knee angle in Bionic Leg (BioLeg.Knee) was 0.4826. Data on the Figure 6 was collected by the number of 50 working cycles including the slandered deviation for the data variance of arm, actual knee, predicted knee and Bionic leg knee as 1.764, 1.882, 1.837 and 1.934 degrees respectively.

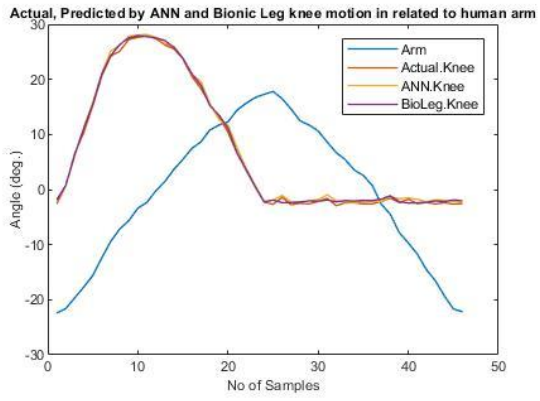


Fig. 6 Terminal Stance and Pre-Swing phase in gait cycle for Bionic Leg, collected data and Predicted Data

V. CONCLUSION

Instead of current bionic leg development for the amputation of leg above the knee joint, this model was developed for the amputation of above knee, in case of, having inactive muscles. Current development of commercial bionic leg unable to detect any data from inactive muscle using EMG. Some others introduced EEG based bionic development to fulfil this problem but using of EEG is not user-friendly due to difficulties of the signal detection and signal conditioning also human bipedal walking is rhythmic activity that control form the CPG (Central Pattern Generator) of the body. This project expected to develop the Artificial Neural Network model to rehabilitate lower-limb gait pattern using bionic leg. Artificial Neural Network model was intelligent to predict the lower-limb gait pattern for the person's arm swinging pattern. In this attempt, Introduced the data analysis part to train the Artificial Neural Network model by using effective sensor modules and expressed proper data types as the features of the model. At the end of this research project, bionic leg prototype was developed using the Artificial Neural network (ANN) model and knee joint control algorithm. to control knee joint according to arm swinging pattern. According to the results, bionic leg has a capability to be predict the corresponding knee joint angle using human arm swinging pattern around execution delay 13 milliseconds. As a result of, gait cycle with bionic leg may have a time delay deviation by 1.21% with healthy human

ACKNOWLEDGMENT

My heartfelt thanks goes to Soft Robotics Research Group of, The Open University of Sri Lanka that operate under the guidance of Principle Investigator and Co-Investigator and thankfully mentioned Samith Wijesinghe, Mininidu Sheheran, Nadun Kaluarachchi and Vijayanadan Rishan for their valuable assistance in this research projects.

REFERENCES

[1] 97.170 - Body care equipment. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:8549:-2:ed-1:v1:en. [Accessed: 01-Aug-2018].

[2] M. Awad, A. Abouhossein, A. Dehghani-Sanij, R. Richardson, D. Moser, S. Zahedi and D. Bradley, "Towards a Smart Semi-Active Prosthetic Leg: Preliminary Assessment and Testing", IFAC-PapersOnLine, vol. 49, no. 21, pp. 170-176, 2016.

[3] "Prosthetics: A simple introduction to artificial limbs", Explain that Stuff, 2018. [Online]. Available: http://www.explainthatstuff.com/prosthetic-artificial-limbs.html. [Accessed: 05- Jan- 2018].

[4] Y. Zeng, Design And Testing Of A Passive Prosthetic Ankle With Mechanical Performance Similar To That Of A Natural Ankle. Milwaukee, WI: E-Publications@Marquette, 2013.

[5] J. Frankenfield, "Artificial Neural Network (ANN)," Investopedia, 28-Aug-2020. [Online]. Available: https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp. [Accessed: 09-Jun-2020].

[6] "A graphic showing Lower Extremity Muscles: Lower limb, Lower limb muscles, Leg anatomy," Pinterest. [Online]. Available: https://www.pinterest.com/pin/448319337885212647/?autologin=true. [Accessed: 06-Oct-2018].

[7] "Amputations," Disabled American Veterans - Do you qualify for benefits? [Online]. Available: http://www.militarydisabilitymadeeasy.com/amputations.html. [Accessed: 10-Mar-2018].

[8] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: detection, processing, classification and applications," Biological Procedures Online, vol. 8, no. 1, pp. 11–35, 2006.

[9] J. Park, "Synthesis of natural arm swing motion in human bipedal walking," Journal of Biomechanics, vol. 41, no. 7, pp. 1417–1426, 2008.

[10] P. Meyns, S. M. Bruijn, and J. Duysens, "The how and why of arm swing during human walking," Gait & Posture, vol. 38, no. 4, pp. 555–562, 2013.

[11] E. P. Zehr and J. Duysens, "Regulation of Arm and Leg Movement during Human Locomotion," The Neuroscientist, vol. 10, no. 4, pp. 347–361, 2004.

[12] P. Martinsen, "Three Methods to Filter Noisy Arduino Measurements," MegunoLink, 23-Nov-2018. [Online]. Available: https://www.megunolink.com/articles/coding/3-methods-filter-noisy-arduino-measurements/. [Accessed: 01-Aug-2018].

[13] H. Xie, Z. Li, and F. Li, "Bionics Design of Artificial Leg and Experimental Modeling Research of Pneumatic

Artificial Muscles," Journal of Robotics, vol. 2020, pp. 1–11, 2020.

[14] J. E. Downey, J. Brooks, and S. J. Bensmaia, "Artificial sensory feedback for bionic hands," Intelligent Biomechatronics in Neurorehabilitation, pp. 131–145, 2020.

[15] X. Liu, Z. Zhou, J. Mai, and Q. Wang, "Real-time mode recognition based assistive torque control of bionic knee exoskeleton for sit-to-stand and stand-to-sit transitions," Robotics and Autonomous Systems, vol. 119, pp. 209–220, 2019.

[16] J. L. Maher, C. B. Baunsgaard, J. V. Gerven, A. E. Palermo, F. Biering-Sorensen, A. Mendez, R. W. Irwin, and M. S. Nash, "Differences in Acute Metabolic Responses to Bionic and Nonbionic Ambulation in Spinal Cord Injured Humans and Controls," Archives of Physical Medicine and Rehabilitation, vol. 101, no. 1, pp. 121–129, 2020.

[17] R. Kurniawan, A. S. Setiyoko, M. A. Jamiin, B. Herijono, F. I. P. Ramadhan, W. Astuti, F. Y. Rizaldi, M. B. Aji, M. Syai'in, Adianto, R. Indarti, and N. Rinanto, "Electric Bionic Legs Used Gyroscope And Accelerometer With Fuzzy Method," 2019 International Symposium on Electronics and Smart Devices (ISESD), 2019.

[18] H. Xie, G. Li, and F. Li, "Modeling of Magnetorheological Damper and Fuzzy PID Control of Intelligent Bionic Leg with Meniscus," 2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA), 2019.

[19] B. Hu, A. M. Simon, and L. Hargrove, "Deep Generative Models With Data Augmentation to Learn Robust Representations of Movement Intention for Powered Leg Prostheses," IEEE Transactions on Medical Robotics and Bionics, vol. 1, no. 4, pp. 267–278, 2019.

[20] S. Das, D. N. Das, and B. Neogi, "Design Approach for Artificial Human Ankle Movement," 2019 Devices for Integrated Circuit (DevIC), 2019.

[21] W. Ling, G. Yu, and Z. Li, "Lower Limb Exercise Rehabilitation Assessment Based on Artificial Intelligence and Medical Big Data," IEEE Access, vol. 7, pp. 126787–126798, 2019.

[22] J. Zhang, S. Liu, Y. Zhang, and R. Zhu, "A Method to Extract Motion Velocities of Limb and Trunk in Human Locomotion," IEEE Access, pp. 1–1, 2020.

[23] A. Alamdari and V. Krovi, "A Review of Computational Musculoskeletal Analysis of Human Lower Extremities," Human Modelling for Bio-Inspired Robotics, pp. 37–73, 2017.

Multi Agent System for Forex Trading

Dewmal Handapangoda[1], Asoka Karunananda[2]
Department of Computational Mathematics
[1,2] University of Moratuwa, SRI LANKA
Email: [1] dewmal@syigen.com, [2] asokakaru@uom.lk

***Abstract*— Foreign exchange market (Forex) is a global marketplace which trades in currencies. This market is more distributed, decentralized, disturbed, and disorganized over the servers on the Internet than the stock market. Due to its very nature, individual investors or experienced traders find it difficult to access, collect, filter, and analyze information to draw meaningful decisions on Forex. Many existing Forex trading solutions use only one of quantitative facts or qualitative facts to reduce the complexity in data analysis. Having evident the potential of Multi Agent Systems (MAS) technology to model complex systems, we have come up with a MAS solution, ForexMA, which builds the mutual influence of quantitative and qualitative facts into decision making. ForexMA is designed with multiple Agents which access, collect, filter, and analyze the qualitative and quantitative information from multiple sources. ForexMA has been tested against its predictions and actual known results and found that performance of ForexMA is above the performance of decision made by human expert traders. More importantly, ForexMA can work on high frequency time frames and generate solutions in few seconds, whereas human expert traders work on low frequency timeframe and take few hours to generate a solution.**

Keywords—Stock Market, Forex Trading, Artificial Intelligence, Multi Agent Systems

I. INTRODUCTION

Forex is the global Foreign Exchange Market where trading is done in currencies of different countries with 5.1 trillion US dollars turnover daily [1]. It is the largest and most liquid asset market in the world. Forex assets are influenced by many factors including news of global political and economic situations, and traders' patterns of price fluctuations. In this sense, any successful trader must access the above information and do an analysis to make an effective decision. However, this analysis is a tedious task due to the inherent complexity of the Forex environment, where the information is distributed, decentralized, disturbed, distorted, and does not provide equal access to all traders. It should be noted that trading in the Forex environment is more complex than Stock Market trading where the trading assets are shares of companies which may not be diversely influenced by other forces [2], [3]. Apart from level of complexity, dynamics of both Forex and Stock Markets suffer from accessibility to the relevant information and analysis of such information.

In view of that numerous researches have been conducted to offer computer-based solutions for Forex trading and Stock Market trading. However, most of these solutions use statistical techniques and computer-based information systems technologies such as MIS. In the recent past, Artificial Intelligence techniques have also been used to model trading in markets. Among others, Multi Agent Systems (MAS) Technology has shown promising results in modelling complex systems such as Forex environments. Next, we discuss selected MAS solutions for trading in Forex and Stock trading.

Davis, Lou and Liu [4] have developed a framework for Multi Agent Systems for stock trading. This solution has been built with agents for gathering and integrating information from diverse sources and enabling decision making. Multi Agent for Forex Trading has been developed by Rui Pedro Barbosa and Orlando Belo [5] and defined six agents to work on six different assets. In general, higher the number of agents in the MAS better the performance. Vivien Delage and Christian Brandlhuber [6] have developed a MAS for understanding the behavior of Forex Market. This research has offered a facility for simulating the Forex market, which allows do experiments Forex trading experiments under different conditions. Xiaorong Chen and Shozo Tokinaga [7] have also developed a Multi Agent based artificial stock market to understand stock market behaviors. Their findings indicate consideration of multiple facts with price changes in assets in the Forex Stock Market.

MAS solution by R. Barbosa and O. Belo [8] automates the analysis of candle patterns, which is a tedious task. Here the candles are data stand for quantitative facts in forex market, and their analysis is an expert task.

Abdullah, Rahaman, and Rahman [9] developed a MAS solution for modelling stock market predictions based on qualitative data. Natural Language processing [10] techniques in Artificial Intelligence have also been used to analyze qualitative news data in social media.

Lee [11] has developed a Forex Advisor hybrid system with MAS and radial basis-function recurrent network (HRBFN). Being a hybrid system, this solution enables the analysis of both qualitative and quantitative facts in the Forex trading. Smart Agent solution which was developed by Alrefaie, Hamouda, and Ramadan [12] used Neuro-Fuzzy Inference, Genetic Programing with Agent technology for their Forex solution.

By Considering the above facts, the power of MAS technology is undisputed to model Forex and Stock trading markets. More importantly, MAS as an AI technology can bring other technologies such as Artificial Neural Networks, Genetic Algorithms, Fuzzy Logic, Natural Language Processing together for empowering Agent's capacity to reach smart solutions. However, due to inherent complex nature of the Forex trading, predicting Forex market dynamics with a higher level of accuracy remains as a research challenge.

As such, this paper presents our research in developing a MAS solution for Forex trading. This approach is different from other solutions as we define Agents to work with qualitative and quantitative data as two distinct sources of information, and agents operating on two different sources deliberate to achieve the final conclusions. As such, we build mutual influence of qualitative and quantitative into our MAS solution. Note that many Forex/Stock solutions have used Artificial Intelligence techniques to model only one qualitative (news) or quantitative (price information) aspects of Forex/Stock trading.

The rest of the paper is organized as follows. Section 2 presents the design of MAS solution, ForexMA. Section 3 is on implementation of ForexMA. Section 4 describes the evaluation of ForexMA. Finally, Section 5 concludes the research with a note on further work.

II. MAS Solution for Forex Trading – ForexMA

In general, depending on how we define agents and their interaction, a MAS solution for a given problem can be modelled in multiple ways. This is analogous to why two different traders could give different recommendations based on the same source of information. Strictly speaking, ForexMA is designed to mimic to the role of an expert trader in the Forex trading.

For ForexMA, we recognize two kinds of facts in Forex trading as Qualitative facts and Quantitative facts, which determine influence from news related information and trading patterns to decision making. Note that sources such as news, opinions, and discussions are treated as qualitative facts. On the other hand, Forex market assets price levels are considered under quantitative facts, which appear as numeric values. More importantly, we implement the mutual influence between qualitative and quantitative facts, without disregarding effect of change in one kind of facts on the other kind of facts.

To deal with qualitative facts and quantitative facts, we have developed several software agents as discussed below. Note that in the backend, these agents may use various statistical techniques and AI techniques such as Artificial Neural Networks to collect, analyze and interpret the information of the Forex market. A discussion on these back-end technologies are beyond the scope of this paper, instead we are interested in modeling interaction among sources through the MAS technology. It should be noted that MAS technology offers facility for incorporating new sources of information, increasing/decreasing the number of active agents when needs arise. Therefore, MAS solutions can be developed as evolvable software.

A. Architecture of ForexMA

Top level architecture of ForexMA is given in Figure 1. Note that Quantitative facts and Qualitative facts are the inputs for the solution and decided action would be the decision generated by the MAS. The MAS comprises three agents, namely, Facts Analyzing Agent, Decision Agent, and Performance Analyzing Agent.

Given two kinds of facts, when Facts Analyzing Agents (FA) finds patterns in the facts, then the patterns are passed to the Decision Agent (DA). Consequently, DAs generate possible decisions with their risk levels. Having deliberated on the possible decisions, an action is passed to what we called Forex trading system, which is the interface for brokers or traders to interact with the solution.

Note that, at once a decision generated by the MAS may not be acceptable to a trader, which results in sending a feedback to Performance Analyzing Agent (PA). Up on the feedback, PA, FA and DA deliberate again and generate a new decision. This shows an amazing feature of MAS technology where a small change in one Agent could lead to emergence of a new solution. More importantly, mutual influence of qualitative and quantitative aspects of the Forex environment could make a significant change in the decision. Just like in a group discussion by humans, in MAS, better solutions are generated in subsequent cycles of deliberations. Let us discuss roles of FA, DA, and FA separately.

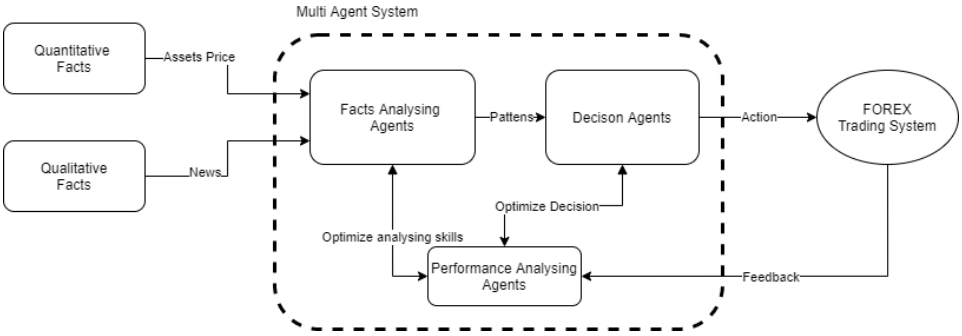


Fig. 6 High Level Architectural Design of proposed Multi Agent System

B. Facts Analyzing Agents

Three types of agents, namely, quantitative pattern analyzing agent, qualitative pattern analyzing agent, pattern analyzing coordinator agent have been developed to implement FA. These agents and their interactions are shown in Figure 2.

Figure 2 shows quantitative pattern agent and qualitative pattern agent receiving two types of facts and identification of quantitative and qualitative patterns. These patterns are sent to pattern analyzing agents to determine the pattern and to release the selected pattern. Note that two types of patterns should be analyzed separately and then need to identify the false patterns by

creating a discussion between both agents. If both agents give the same suggestion, there is no further discussion but if they are different the Pattern Analyzing Coordinator Agent must participate in deliberation. Those quantitative pattern analyzing agents and qualitative pattern analyzing agents are designed to use Artificial Neural Networks to achieve their pattern analyzing power.

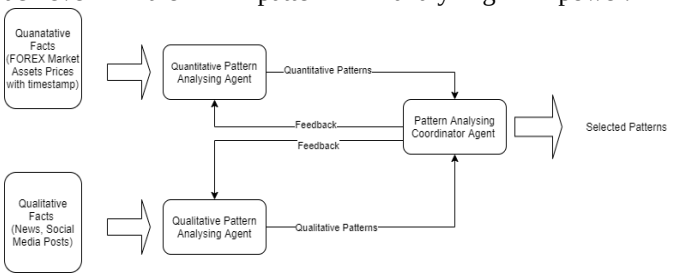


Fig. 7 Types of Facts Analyzing Agent

C. Decision Agent & Performance Analyzing Agent

Since DA and PA cannot operate independently, they are discussed together. To perform a trade in the Forex trading environment, the trading-instructions must be provided. In our system DA provides the necessary trading-instructions. Based on these instructions, DA deliberates with PA to generate Actions relevant to the pattern.

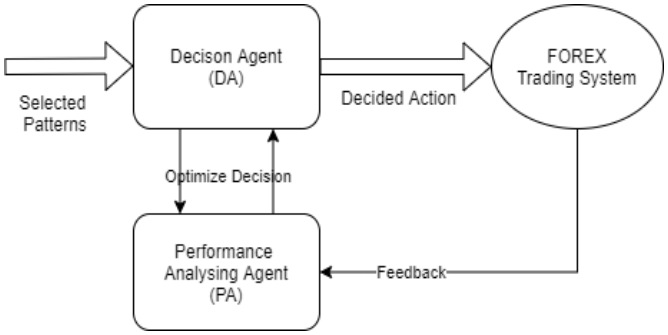


Fig. 3 Interaction between DA and PA

When DA receives patterns from FA, DA generates a decision by analyzing the patterns. Then it sends the decision to PA to get confidence value based on similar decisions in the past. After receiving the confidence from PA, DA evaluates its decision based on confidence value. If the DA is satisfied with the decision it generates an action (trading-instruction) and sends it to the Forex trading system. If it is not satisfied with the decision it rejects the decision and waits for the next pattern. If it is partially satisfied with the decision it restarts the analyzing process to optimize the decision.

The optimization of a decision is done by generating another decision based on the same pattern set. Then DA & PA perform the same analysis process explained earlier. This iterative optimization process leads to generating more accurate decisions.

III. The Implementation of the ForexMA

This section briefly describes the implementation of the design of MAS solution stated above. ForexMA has been implemented as a MAS solution on python and Redis based agent platform. Redis creates a pub-sub channel for ForexMA agents to have a Real Time distributed message passing system. Because of that agents can be run in different locations based on their requirements of hardware. Like Qualitative FA need GPU support.

A. Implementation of Agents

ForexMA is designed to achieve its end goal by performing three types of agents and those agents are specialized for different types of tasks. However, implementing these agents are not different from each other's core features. Figure 4 is showing the way of implementation of Agent in ForexMA.

The implementation of Agent mainly focuses on improving the communication power of each agent according to their tasks in the designed section. Therefore, each agent has their own communication channel, for example, DA has its own communication channel which is named as Decision-Agent. Once that channel receives a message from another agent Message Receiver start call message driven methods. After that agent can perform any action based on that message.

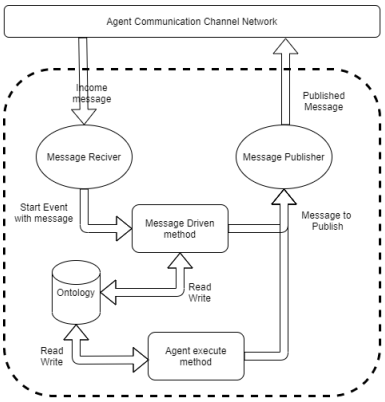


Fig. 8 General Implementation of Agents

Sometimes agents need to perform tasks at the beginning of the agent start to work. For that purpose, there is a method called execute to such a task.

Here is an example of agent skeleton code.

```
class QualitativeAnalyticalAgent:
 name = "QualitativeAnalyticalAgent"

  # Execute with start of agent
      async def start(self):
                pass
   #Execute on start of agent
   async def accept_message(self,
       agent, message):
                pass
 #Execute before stop of agent
    async def stop(self, *args,
          **kwargs):
                pass
 #Execute after start function
  async def execute(self, *args,
          **kwargs):
     await self.publish("AgentTwo",
                {
```

```
            "message": "Qualitative
    Analytical Agent Found a Pattern"
                    })
```

B. Implementation of FA

The design of the ForexMA supports two types of data such as qualitative facts and quantitative facts. To achieve that goal, We need to implement two types of agents such as Qualitative Facts Analyzing Agent and Quantitative Facts Analyzing Agent. Both agents are work same but two different kinds of data types consider as the inputs. Quantitative FA uses a rule based Expert System to identify the price action patterns from received price data. In contrast, Qualitative FA uses Artificial Neural Network to perform the analyzing the sentimental value of given qualitative facts such as news, social media posts. Figure 5 shows implementation of FA. Accordingly, FA use message driven execution method to perform facts analysis.

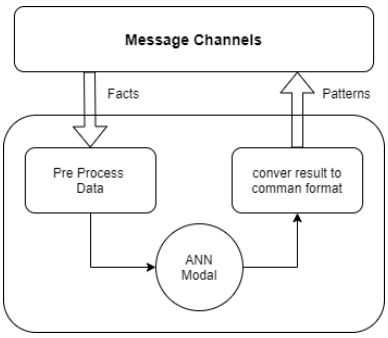


Fig. 9 FA implementation

C. Implementation of DA

After the patterns received from Fact Analyzing Agent, DA need to inference those patterns. Because of that, the implementation of DA needs to integrate with Rule Based Expert System (ES). To the implementation of ES should have to use an external python library which is called 'Experta'. Figure 6 shows the implementation of the DA.

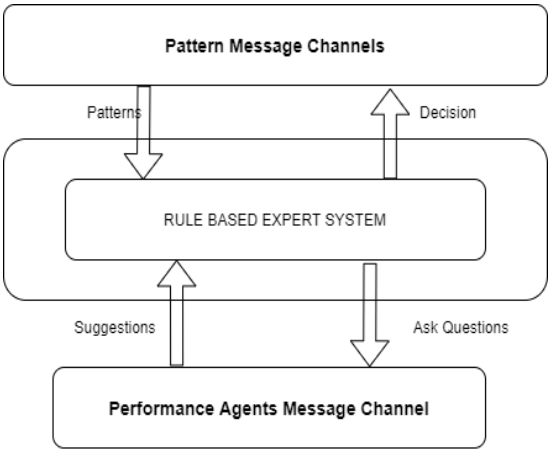


Fig. 10 Implementation for decision agent

According to Figure 6, DA's message channel is getting messages from both PA and FA. The flow of the status is managed by DA's ontology which is implemented using SQLight.

D. Implementation of PA

PA plays a key role in our MAS solution by receiving feedback from Forex Trading System and using that feedback, PA keeps a record about all other agent's performance based on their actions against the received feedbacks trading instructions. For analyzing and optimization purposes, PA has been implemented as an expert system. Also, that implementation support communicating with DA for having a performance analysis of DA's decisions.

E. Implementation of agent's communication

Each Agent has its own communication channel. The implementation of the channel communication, we used the subscriber design pattern. Figure 7 shows the implementation of DA's message channel, which is the same for all other agents.

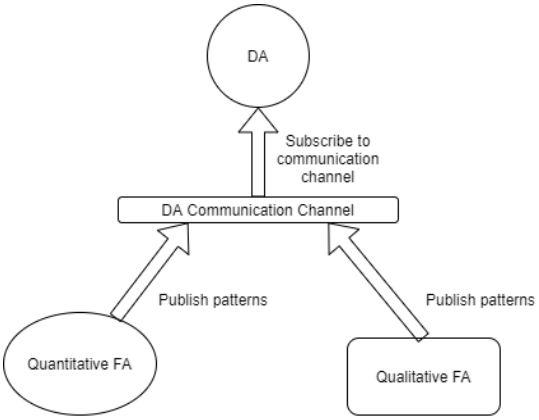


Fig. 11 Implementation of Agents communication Channel

According to Figure 7, DA has its own message channel which is called DA Communication Channel. DA subscribes to that channel and any other agents can publish messages to that channel using the following line of python code.

```
self.publish("DecisionAgent", {"data":
                {} })
```

Once a message is arrived at the DA's Message Channel, the following method is called by the agent platform.

```
async def accept_message(self, agent,
                message):
                    ...
```

Parameters of these methods, agent stands for sender Agent's Name and message brings the sender's data to the receiving agents. Using those two details Agents can perform its tasks.

F. Agents Message passing flow

According to the implementation of message channels, each agent has its own message passing channel as shown in Figure 8.

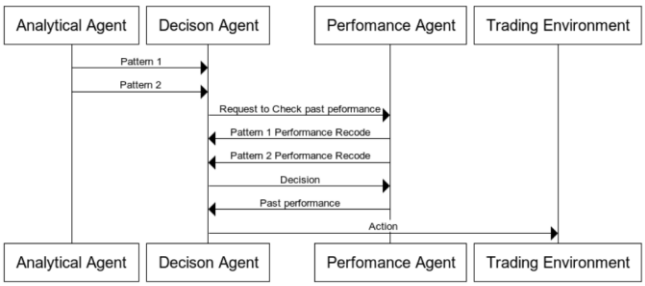


Fig. 12 Agents Message passing flow

Figure 8 shows an example of message flow between FA, DA, PA, and Forex Trading systems. Note that, there is no predefined order of passing messages by agents, yet any agent can pass a message when a message is ready.

IV. Evaluation of ForexMA

To discuss evaluation of ForexMA, the first reader should have an idea of how Forex trading works. Forex markets allow traders to gain profit by investing their money in multiple trading types such as day trading, swing trading and position trading. Regarding an asset, traders need to identify the trend before making any decision. A trend shows the direction in which the selected asset is heading. For a given trend, traders need to identify its measurable fact such as variation of the price. ForexMA is designed to generate actions with direction of trend and expected amount of price variation.

Forex market assets price frequently change because of the market trend. Traders try to understand the market trend by analyzing qualitative facts such as news, social media posts. Once they identify the trend, they can calculate the expected price variation due to the trend. Note that frequency of price change happens more than twice a second. Because of the high frequency of price variation, we must consider the price change for a time window.

For a given time window, we are interested only in a few key price values. They are, value at the opening of the time window (OPEN), value at the end of the time window (CLOSE), highest value in the time window (HIGH) and lowest value in the time window (LOW).

As per above discussion, the evaluation of performance of ForexMA should validate the combined prediction power of *trend identification* and *price variation*. Therefore, reward functions have been used to calculate a combined reward value.

A. Inputs of the ForexMA

ForexMA needs quantitative facts and qualitative facts as inputs. We have used price variations, as quantitative facts of EUR-USD between the 2019-11-27 and 2020-10-15. For the same time duration, we have also used textual data such as news, social media posts which are related to EUR-USD, as qualitative data.

Table 1 shows a sample of quantitative data collected in 5 minutes intervals for three days. Here the volume of data is about 65000 records. Note that when human expert traders analyze the Forex market, normally by collecting data only every 4-8 hours, this results in reducing the accuracy of prediction. As such making a time frame as 5 minutes has a big contribution to improve accuracy of making decisions.

Table 1 Quantitative Input for ForexMA

Timestamp	Open	High	Low	Close
2020.10.08 14:00	1.65	1.658	1.649	1.647
2020.10.08 14:05	1.647	1.652	1.642	1.643
2020.10.08 14:10	1.643	1.67	1.643	1.66
2020.10.08 14:15	1.66	1.67	1.658	1.662

Table 2 Qualitative Inputs for ForexMA

Timestamp	Text
2020.10.08 12:00	Paris hospitals postpone non-essential operations – as it happened
2020.10.08 14:48	Five Greek islands added to England's quarantine-free list
2020.10.08 15:23	Covid drug given to Trump developed using cells derived from…
2020.10.08 18:16	New Zealand National party leader yearns for the star treatment

In addition to quantitative inputs, ForexMA receives qualitative inputs pertaining to news. We have used data crawlers to collect news from news websites and used their title to generate ForexMA outputs. ForexMA used those title and identifies the sentimental value of them and that is helping to generate the output action. Qualitative data does not have a unique period, Once update the original news source crawlers will collect the data and feed them to the ForexMA. Table 2 shows few samples of qualitative data.

B. Output of ForexMA

Input Figures to ForexMA (Table 1, Table 2) cannot be comprehended by the trader to make decisions, but the trader is interested in the output of ForexMA. Table 3 shows the outputs of the ForexMA as Predicted Action (PA), Predicted Price Variation (PPV) and End Timestamp. Note that these figures can occur at any time. There is no relationship between Timestamps in Table 1 and Table 3. It should be noted that figures in Table 3 have been generated by ForexMA, after deliberation between quantitative and qualitative facts shown in Table 1 and Table 2 by the agents in the MAS solution. These PA and PPV are used by Forex traders for decision making.

Table 3 Output of ForexMA

Timestamp	End Timestamp	PA	PPV
2020.10.08 15:00	2020.10.08 15:10	BUY	20
2020.10.08 15:15	2020.10.08 15:25	SELL	5
2020.10.08 15:30	2020.10.08 15:40	BUY	115
2020.10.08 15:55	2020.10.08 15:55	BUY	90

C. Performance test for ForexMA

For the evaluation process, we need to compare performance from known data and output created by ForexMA. For this purpose, many literatures have used the formula [13] to compute two parameters, namely, Actual Action (AA) and Actual Price Variation (APV). These AA and APV are computed for known data and ForexMA

generated solutions. Figure 9 shows the algorithm for calculating AA and APV.

Algorithm parameters: ETP, STP
Algorithm outputs: AA, APV

Initialize V, AA, APV
Calculate $V = ETP - STP$
If $V > 0$:
  Asign $AA =' SELL'$
else:
  Asign $AA =' BUY'$
end if
  Assign $APV = |V|$
return AA, APV

Fig. 13 Actual Action and Actual Price Variation Calculation Algorithm

Where,
ETP – Assets Price at the give end point (End Timestamp)
STP – Assets Price at the given start point (Start Timestamp)
AA – Actual Action which supposed to come with output of ForexMA
APV – Actual Price Variation which supposed to come with output of ForexMA

Each output of ForexMA must compare with the actual status of the market. For that purpose, we used the above described scenario to calculate AA and APV. Table 4 shows samples rows of actual action and actual price variation.

Table 4 Calculated Actual Outputs

Timestamp	AA	APV
2020.10.08 15:00	SELL	12
2020.10.08 15:15	SELL	15
2020.10.08 15:30	BUY	75
2020.10.08 15:55	BUY	150

We calculate reward for each action. If AA is not equal to PA we give reward as 0, and if AA is equal to PV we calculate the reward [14] using formula (1).

$$\text{Reward} = 1 - | \text{APV} - \text{PPV} | / | (\text{APV} - \text{PPV})/2 | . \quad (1)$$

Where,
APV - Actual Price Variation
PPV - Predicted Price Variation

Once we calculate each action's reward value, we use the formula (2) to calculate the performance of the sample. Note that performance is a percentage value and highest is the best.

$$\text{Performance} = (\text{Sum of all Rewards} / \text{Number of actions}) \times 100 . \quad (2)$$

D. Evaluation of Test Results

We have used 4 different known datasets considering USD-JPY, GBP-USD, AUD-USD and EUR-USD of Forex trading and calculated performance as per formulae (1). Each of these data sets consists of 65000 records. Table 5 shows performance of ForexMA related to each decision.

According to Table 5, the average performance of ForexMA is 57.11%. This is higher than the performance of expert traders, which is average to 55% [15], [16]. It should also be noted that ForexMA can generate the solution with the above accuracy in a few seconds. However, in general, human traders takes few hours for generating a decision.

Table 5 Performance analysis details for each sample

Sample	Time duration	Asset	Performance %
1	2019.06.08 2020.05.11	USD-JPY	58.23%
2	2019.06.15 2020.05.18	GBP-USD	55.85%
3	2019.06.15 2020.05.18	AUD-USD	52.63%
4	2019.09.22 2020.08.25	EUR-USD	63.38%
5	2019.09.22 2020.08.25	AUD-USD	52.63%
6	2019.09.22 2020.08.25	GBP-USD	56.15%
7	2019.11.27 2020.10.15	USD-JPY	59.03%
8	2019.11.27 2020.10.15	AUD-USD	52.63%
9	2019.11.27 2020.10.15	GBP-USD	56.35%
10	2019.11.27 2020.10.15	EUR-USD	64.18%

V. CONCLUSION AND FURTHER WORK

Forex is the largest and most liquidated asset market in the world by reason of the involvement of multiple factors. This paper narrates design, implementation, and evaluation of MAS solution, ForexMA, for Forex trading. ForexMA takes into consideration qualitative and qualitative facts to make decisions about Forex trading. The overall system developed on Python and Redis based MAS platform. This agent solution is designed to explore the power of several AI technologies, including Artificial Neural Network and Expert System. ForexMA is significantly different from other solutions for Forex trading, where they consider only one of the qualitative or quantitative facts related to Forex trading. Our ForexMA implements the mutual influences of qualitative and quantitative measures of Forex trading.

According to results of evaluation, ForexMA has shown 57% performance. This value is above the average performance of expert human traders, which is 55%. It should also be noted that ForexMA generates a decision in few seconds, while the human traders take few hours for the same but with lesser accuracy in performance. Furthermore, ForexMA can work on high frequency time frames such as 1-minute, 5-minute durations to archive higher accuracy. However most human traders do not perform well in such data frames. They generally work on 4-hour to 8-hour time durations, in which case accuracy of predictions goes down.

Further work of this research has been identified as improving agents with more rules used by expert traders when they make decisions. We should also ensure that

ForexMA access the reliable data sources to get quality data for analysis. When ForexMA deals with qualitative data such as news and information coming from social media, this matter is of utmost importance. We also intend to expand ForexMA with a mobile interface.

REFERENCES

[1] M. Ozturk, I. H. Toroslu, and G. Fidan, "Heuristic based trading system on Forex data using technical indicator rules," Appl. Soft Comput., vol. 43, pp. 170–186, Jun. 2016, doi: 10.1016/j.asoc.2016.01.048.

[2] N.-F. Chen, R. Roll, and S. A. Ross, "Economic Forces and the Stock Market," J. Bus., vol. 59, no. 3, p. 383, Jan. 1986, doi: 10.1086/296344.

[3] J. Shanken and M. I. Weinstein, "Economic forces and the stock market revisited," J. Empir. Finance, vol. 13, no. 2, pp. 129–144, Mar. 2006, doi: 10.1016/j.jempfin.2005.09.001.

[4] D. Davis, Y. Luo, and K. Liu, "A Multi-Agent Framework for Stock Trading," p. 7.

[5] R. P. Barbosa and O. Belo, "Multi-Agent Forex Trading System," in Agent and Multi-agent Technology for Internet and Enterprise Systems, A. Hãkansson, R. Hartung, and N. T. Nguyen, Eds. Berlin, Heidelberg: Springer, 2010, pp. 91–118.

[6] DELAGE, C. BRANDLHUBER, K. TUYLS, AND G. WEISS, "MULTI-AGENT BASED SIMULATION OF FOREX EXCHANGE MARKET," JAN. 2011.

[7] X. Chen and S. Tokinaga, "MULTI-AGENT-BASED MODELING OF ARTIFICIAL STOCK MARKETS BY USING THE CO-EVOLUTIONARY GP APPROACH," J. Oper. Res. Soc. Jpn., vol. 47, no. 3, pp. 163–181, 2004, doi: 10.15807/jorsj.47.163.

[8] K. H. Lee and G. S. Jo, "Expert system for predicting stock market timing using a candlestick chart," Expert Syst. Appl., vol. 16, no. 4, pp. 357–364, May 1999, doi: 10.1016/S0957-4174(99)00011-1.

[9] S. S. Abdullah, M. S. Rahaman, and M. S. Rahman, "Analysis of stock market using text mining and natural language processing," in 2013 International Conference on Informatics, Electronics and Vision (ICIEV), May 2013, pp. 1–6, doi: 10.1109/ICIEV.2013.6572673.

[10] F. Ozcan, "Exchange Rate Prediction from Twitter's Trending Topics," p. 46.

[11] R. S. T. Lee, "iJADE stock advisor: an intelligent agent based stock prediction system using hybrid RBF recurrent network," IEEE Trans. Syst. Man Cybern. - Part Syst. Hum., vol. 34, no. 3, pp. 421–428, May 2004, doi: 10.1109/TSMCA.2004.824871.

[12] M. T. Alrefaie, D. Hamouda, and R. Ramadan, "A Smart Agent to Trade and Predict Foreign Exchange Market," Comput. Intell. Eng. Solut. CIES 2013 IEEE Symp. On, Apr. 2013, doi: 10.1109/CIES.2013.6611741.

[13] S. Papadamou and S. Tsopoglou, "Investigating the profitability of technical analysis systems on foreign exchange markets," Manag. Finance, vol. 27, no. 8, pp. 63–78, Aug. 2001, doi: 10.1108/03074350110767349.

[14] J. Carapuço, R. Neves, and N. Horta, "Reinforcement learning applied to Forex trading," Appl. Soft Comput., vol. 73, Sep. 2018, doi: 10.1016/j.asoc.2018.09.017.

[15] M. Islam, E. Hossain, A. Rahman, M. Hossain, and K. Andersson, "A Review on Recent Advancements in FOREX Currency Prediction," Algorithms, vol. 13, p. 186, Jul. 2020, doi: 10.3390/a13080186.

[16] C. Gallo and A. Fratello, "The Forex Market in Practice: A Computing Approach for Automated Trading Strategies," Int. J. Econ. Manag. Sci., vol. 3, Aug. 2014, doi: 10.4172/2162-6359.1000169.

Semi-supervised learning approach to multiclass object detection with obscure and overlapping boundaries

Lahiru R. Rathnayake[1], and Ruwan D. Nawarathna[2]
[1]Postgraduate Institute of Science, University of Peradeniya, Peradeniya 20400, Sri Lanka
[2]Department of Statistics and Computer Science, University of Peradeniya, Peradeniya 20400, Sri Lanka
email: [1] lahrumesh28@gmail.com, [2] ruwand@pdn.ac.lk

***Abstract*— Multiclass object detection has a variety of uses in the field of computer vision and many of the object detection algorithms are implemented for real-world tasks. A common issue in the area of multiclass object detection is the overlapping objects with unclear boundaries. We propose a method to automatically detect objects with obscure and overlapping boundaries in an image by utilizing a fine-tuned object detection model based on YOLOv3 coupled with a semi-supervised learning approach. To test the model, a dataset containing 5000 images of 8 different classes of grocery items was used. Also, we employed a semi-supervised learning technique called pseudo labeling to minimize the time for labeling of the data. We show that pseudo labeling eliminates the need for expensive manual verification and labeling process which minimize requirements for domain experts in many applications. The final model trained with semi-supervised learning archives a mean average precision(mAP) of 0.895 on the test data set.**

Keywords*— Computer Vision, Multiclass Object Detection, Pseudo Labeling, Semi-supervised learning

I. INTRODUCTION

Over the last years, researchers in the field of Computer Vision have relied heavily upon machine learning approaches that focused on the problem of identifying the visual appearance of objects. From traditional image classification problems [1], they shifted to the more challenging tasks of object detection and segmentation [2]. Neural network based multiclass object detection is one of the prominent applications of Computer Vision[3]. It is concerned with finding and locating specific objects in an image. It appears in many real-world situations, such as medical imaging, traffic management systems, face recognition and self-driving cars. Humans can identify and learn real-world objects effortlessly, but lack of human resources makes it difficult and expensive to obtain human labor for object identification. Identifying objects depend on their appearances which fall into two classes. Some objects may be clearly separated from the other objects as shown in Fig. 1(a), which falls into category of individual isolated object detection. In the other class, objects may overlap with other objects and only a part of the object may be seen as displayed in Fig. 1(b) since the object boundaries are overlapped. This creates a much more challenging object identification problem of overlapped and obscured objects.

In almost all object identification and localization applications, it is required to use multiclass object detection techniques. We decided to develop a multiclass overlapped object detection model for a real-world problem of grocery item detection (See Fig. 1). Most supermarkets rely on traditional barcode readers for item identification during checkout. This process takes a lot of time and can result in long queues and dissatisfied customers. During the period of a pandemic such as Covid-19, dealing with these long queues can be problematic. Multiclass object detection method would make the cashiers' checkout process quicker and easier. On many occasions, customers use a basket or a trolley to gather grocery items. Generally, these items are visibly overlapped with other items as shown in Fig. 1(b) where edges and boundaries are unclear.

From the bucket of Deep Learning algorithms, Convolution Neural Networks (CNNs) are commonly used for image classification and object detection[4] . Edge detection is one of the influential concepts in the field of Computer Vision which is the first phase of object detection [5]. Typically, the first few layers of a CNN perform the feature extraction required for object detection including edge detection. There are two categories of object detection methods. The first one is the two-stage detector in which the model proposes a set of regions of

Figure 1 a) Individual objects non-overlapping object boundaries and b) overlapped objects obscure object boundaries.

interest through select search or regional proposal networks and performs classification on the selective regions [6]. This approach is mainly used in R-CNN, Fast R-CNN and Faster R-CNN models [7], [8], [9]. The second one is the one-stage detector that skips region proposals and run detection directly over a dense sampling of possible locations. Examples for this category include AttentionNet, SSD and YOLO models [10], [11], [12]. Drid et al. [13] presents a comprehensive review of detecting overlapping objects from two-stage and one-stage detector models by using PASCAL VOC dataset [14]. They have proposed a model to combine both models to enhance performance. We propose an algorithm to deal with the challenging task of identification of objects with obscure and overlapping boundaries using a CNN-based

multi-object detection model trained with a semi-supervised algorithm [15].

The remainder of the paper is organized as follows. Section 2 presents the background, details of the models and techniques used, and the related work of the study. The proposed model is described in Section 3 and the experimental results after applying the proposed model on a sample data set are summarized and discussed in Section 4. Finally, Section 5 provides concluding remarks and future works.

II. BACKGROUND

A. Machine Learning for Object Detection

There are distinct machine learning algorithms available for object detection tasks. In general, machine learning algorithms can be divided into 3 main types, namely, supervised learning, unsupervised learning, and semi-supervised learning[16]. The majority of the problems in machine learning use supervised learning methods. Supervised learning models in computer vision problems learn from the labeled dataset which contains input images attached with appropriate labels. Unsupervised learning algorithms work with unlabeled data. Semi-supervised learning which is used in this study uses a combination of supervised and unsupervised learning techniques because it works with both labeled and unlabeled data.

Bounding box is one of the widely used image annotation methods for object detection in machine learning[17]. The purpose of adding bounding boxes is to highlight the visible contents of the image as shown in Fig. 2.

Figure 2 Bounding box annotations

The bounding box is a rectangular box that is determined by a point, width, and height according to the pixels in an image. In object detection methods input is an image with one or more objects and output is one or more bounding boxes and class label for each bounding box. Image annotation extends to instance segmentation where object boundaries are highlighted at pixels level. In supervised learning, human input is required to annotate enormous amounts of data manually which can be extremely challenging. Usually, deep learning models require considerably large data sets to make the final model more accurate and robust. It would be such a waste if unlabeled data is not used for creating the object detection model due to the tedious labeling process. To overcome this issue, we propose a technique called Pseudo Labeling.

B. Pseudo Labeling

The technique of using a partially trained model to label unlabeled data falls under the category of Pseudo Labeling. Pseudo labeling method uses a small set of labeled data with a large quantity of unlabeled data to enhance the model accuracy [18]. Before bidding for a pseudo labeling process, it is necessary to ensure that the partially trained model performs well during training and validation. Also, the labeled data should be a proper representation of the full data set. Also, there is a possibility of mislabeling the remaining unlabeled data which may cause an adverse effect on the performance of the model. To overcome this issue only the pseudo-labeled samples of a class that obtain a predicted probability that is greater than a particular threshold value is used. Even though this technique does not completely eliminate the risk of mislabeling, it helps to reduce the burden of mislabeled data.

C. Object detection with Transfer Learning

As previously pointed out in Section 1, the most effective technique for the task of object detection is the use of deep convolutional neural networks (CNNs). However, the training time of CNN models on large data sets can be extremely high. A way to overcome this problem is to use weights from pre-trained models which are generated from computer vision benchmark datasets. ImageNet and Microsoft Common Objects in Context (MS COCO) are the most widely used datasets use for computer vision projects[19], [20]. ImageNet contains more than 14 million images with 22000 visual categories and on the other hand, MS COCO provides an accessible object detection image dataset that contains 91 object types with a total of 2.5 million annotated instances in 320,000 images.

Transfer learning is a method that reuses models trained for a similar problem or a slightly different task by fine-tuning the parameters of the pre-trained model[21]. This approach is highly effective for the feature extraction process of CNNs containing several convolution layers followed by max pooling layers[22]. Basically, CNN layers have weight matrices, which are updated during the training process via the backpropagation algorithm [23]. Typically, these multiple forward and backward iterations of the backpropagation algorithm may lead to a high training time. To build the final model, we can directly apply the weights and the model architecture of a pre-trained model trained on large datasets. From a practical perspective, transfer learning can be achieved through (1) training the entire model from scratch or (2) training some layers and leave others frozen. These two approaches depend on the size of the data set and how similar is the new problem to the problem considered for the pre-trained model. A pre-trained model may not be 100% accurate for every application, but it eliminates the huge effort required to build models from scratch.

D. Faster-RCNN

In contrast to object classification models, the object detection and localization models use a bounding box around the object of interest to locate it within the image. Deep learning techniques like Region-based Convolution

Neural Networks (R-CNN) are developed exactly for that purpose, which use selective search to extract regions for an image. Faster R-CNN is an object detection algorithm that is similar to R-CNN. This algorithm utilizes the Region Proposal Network (RPN) that shares full-image convolutional features with the detection network in a cost-effective manner than R-CNN[24]. Instead of using a selective search algorithm as in R-CNN, Faster R-CNN uses a feature map to identify the region proposals and a separate network to predict the region proposals. These region proposals pass through a fully connected convolution layer with softmax classifier to classify the bounding boxes of the image. Fig. 3 illustrates the Faster-RCNN model.

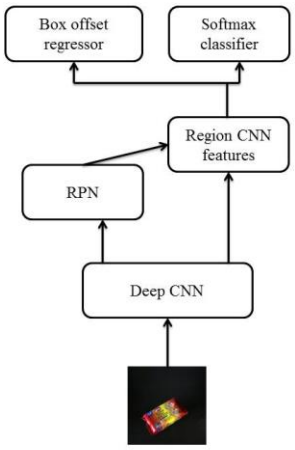


Figure 3 Faster R-CNN model

E. YOLOv3

You Only Look Once or YOLO is a popular algorithm used for object detection and localization [25]. In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes as shown in Fig. 4. The algorithm divides the image into grids and runs the image classification and localization algorithm on each of the grid cells. It predicts N bounding boxes and confidence scores in each grid. The confidence score reflects the accuracy of the bounding box of that class. Bounding boxes having the class confidence above a threshold value are selected and used to locate the object within the image.

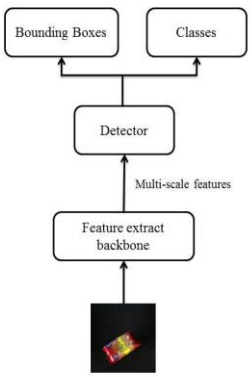


Figure 4 YOLOv3 model

YOLOv3 is an improvement over previous YOLO detection networks [26]. YOLOv3 predicts the coordinates of bounding boxes directly using fully connected layers on top of the convolutional feature extractor. Faster R-CNN object detection model described in Section 2.3 predicts bounding boxes using hand-picked anchor boxes which is somewhat different from YOLOv3 [27]. YOLOv3 uses independent logistic classifiers and binary cross-entropy loss for class prediction.

III. METHOD

In this study, we evaluate the best approach for the detection and localization of overlapping grocery items. This section describes the data collection, the models used, proposed pseudo labeling approach, training models and the evaluation procedure.

A. Data Collection

We captured 5000 images (size of 512x512) of grocery items from 8 different classes, which contain 2000 objects from each class. For the training purposes, we annotated 600 images by using a Visual Object Tagging Tool (VoTT) [28]. Bounding box annotation technique was used as the method to annotate 200 objects from each class. Those classes labeled as “cream cracker”, “sunlight powder”, “milk powder”, “sunlight soap”, “surf excel”, “krisco bites”, “lifebuoy soap”, “signal toothpaste”. The coordinate format of the annotation is defined as (x_{min}, y_{min}, x_{max}, y_{max}). Also, the samples were assigned a label which is a number between 0 and 7. We also create a separate test data set consisting of manually verified samples to reliably evaluate the performance of the models.

B. Models Used

Two models were used in this work to determine which model best suits our task, namely Faster R-CNN with ResNet-101-FPN as backbone architecture (Model 1) and YOLOv3 with a Darknet-53 architecture (Model 2) [29]. We use the PyTorch library to obtain pre-trained models and fine-tunned models [30].

C. Training of the Models

The dataset was split into train sets and validation sets with 90% and 10% of the samples randomly assigned to each set, respectively. We divide the training process into 2 phases. For Training Phase 1, we initialize the models with pretrained weights on the COCO dataset. In that stage, we fixed the backbone architecture with weights and train the classification and regression layers. The size of input images was set to 512x512 and the same hyperparameters were used for the two models. Table 1 shows the hyperparameter values used for the two models.

Table 1 Hyperparameter values used for Model 1 and Model 2

Hyperparameter	Values
Learning rate	4e-4
Training batch size	10
Adam epsilon	le-3
Training epochs	60
Weight decay	0.1

D. Evaluation and Pseudo Labeling

In the field of Data Science, the mean average precision (mAP) is a widely used metric to evaluate the performance of multiclass object detection models [31]. Precision measures how many of the object locations made by the model are actually correct whereas the recall measures how many of the actual object localizations have been predicted by the model. Mean average precision is the average of the area under precision-recall curves (AP) of all classes. Formula 1 shows how mAP is calculated.

$$mAP = \frac{1}{N} \sum_{i=0}^{N} AP_i \qquad (1)$$

We created a manual testing data set to evaluate models. Mean Average Precision (mAP) values of 0.798 and 0.845 were achieved for Faster R-CNN and YOLOv3 models respectively. We observe that both models show satisfactory performance on the testing dataset. Sample detections for non-overlapping grocery items are shown in Fig. 5 in which Fig. 5(a) and Fig. 5(b) are the predictions of YOLOv3 and Faster R-CNN models respectively. Fig. 6 illustrates a couple of sample detections for overlapped and obscured objects. It can be observed that YOLOv3 model (Fig. 6(a)) has performed better in detecting overlapping objects compared to Faster R-CNN model (Fig. 6(b)). Therefore, YOLOv3 model was selected for the pseudo labeling approach (i.e., Training Phase 2).

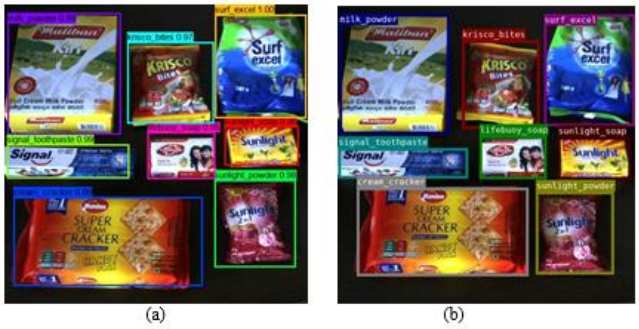


Figure 5 Sample detections for the detection of individual grocery items with non-overlapping boundaries where a) shows the detections of YOLOv3 model and b) shows detections of Faster R-CNN model.

Figure 6 Sample test results for detection of grocery items with overlapping and obscured boundaries where a) shows the detections of YOLOv3 model and b) shows detections of Faster R-CNN model.

.

Fig. 7 illustrates the proposed pseudo labeling process. For this semi-supervised learning method, we used the partially trained YOLOv3 model for pseudo labeling of the remaining 4400 images from the unlabeled data set. In YOLOv3 several convolutional layers are added to the default feature extractor Darknet-53, where the last of these layers predicts the bounding box coordinates, object class and confidence threshold. These 3 output predictions were used for the pseudo labeling approach. Since mislabeling data is known as one of the main drawbacks in the pseudo labeling process, we used a high value of 0.85 as the confidence threshold output. If the model gives the required threshold value, only those object classes and the corresponding bounding box coordinates were used to annotate the images.

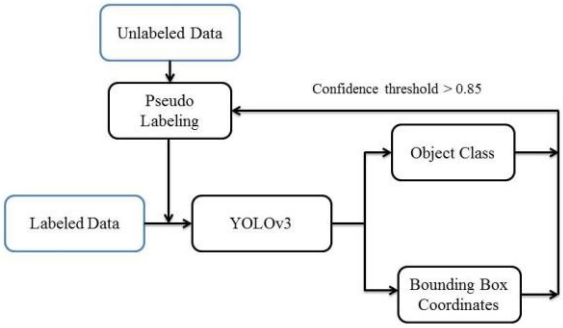


Figure 7 Pseudo labeling process with YOLOv3 model

For Training Process 2 we initialized only the YOLOv3 model and used data augmentation methods to create new training examples[32]. The same hyperparameters with early stopping were applied [33]. Early stopping is a method that allows a model to specify an arbitrarily large number of training epochs and halt training once the model performance stops improving on a hold out validation dataset. We included early stopping to terminate the training if there is no improvement in validation loss after 10 epochs. After Training Process 2 mentioned in Section 3.3, we managed to reach a mAP value of 0.895 for the testing data set. All model evaluations were performed on a single Tesla P40 GPU with use of Pytorch library [33].

IV. Results and Discussion

After Training Process 2 mentioned in Section 3.3, we managed to reach an mAP of 0.895 for the testing data set. Table 2 provides a summary of the mAPs achieved with the models described in Section 3.

Table 2 Testing accuracy for models

Model	mAP
Model 1 (Faster R-CNN with ResNet-101-FPN)	0.798
Model 2 (YOLOv3 with a Darknet-53)	0.845
Model 3 (YOLOv3 with a Darknet-53 + Semi-supervised Pseudo Labeling)	0.895

Comparing the performance of the three models on the test data set, Model 3 (YOLOv3 models with semi-supervised learning) outperforms Model 1 and Model 2. Fig. 8 and Fig. 9 depict the performance of each individual class of Model 2 and Model 3 of each individual class. It can be clearly seen that Model 3 shows better performance for each class compared to Model 2. This is expected since the size of the training data has been increased because of the pseudo labeling process. With the semi-supervised pseudo labeling approach, overall mAP is increased up to 0.895 from 0.845. Also, the average precision of "milk powder", "cream cracker", "sunlight soap", "signal toothpaste" and "sunlight powder" classes are increased. Additionally, there was a significant rise in the average precision (from 0.47 to 0.68) of the "sunlight powder" class. Furthermore, both Model 2 and Model 3 showed a good performance on detecting "surf excel", "milk powder", "cream cracker" and "krisco bites" classes. This is particularly interesting since the primary difference between the two models is the use of pseudo labeling. Since YOLOv3 model is a fast real-time object detector we managed to achieve 30ms of inference time per image detection on Tesla P40 GPU for both of these models. Therefore, the application can be developed for real-world scenarios. However, further work is necessary to increase the average precision of other classes.

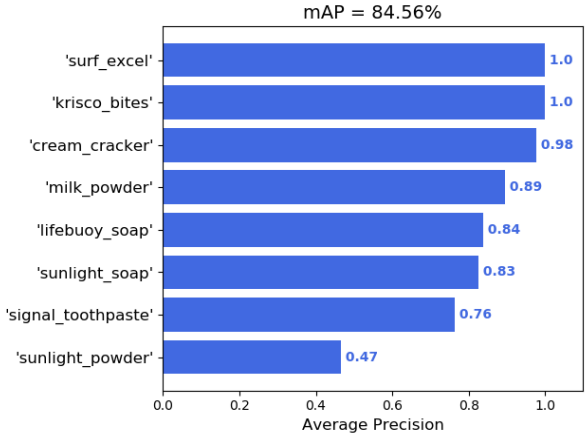


Figure 8 Average precision graph for Model 2 (YOLOv3 with a Darknet-53)

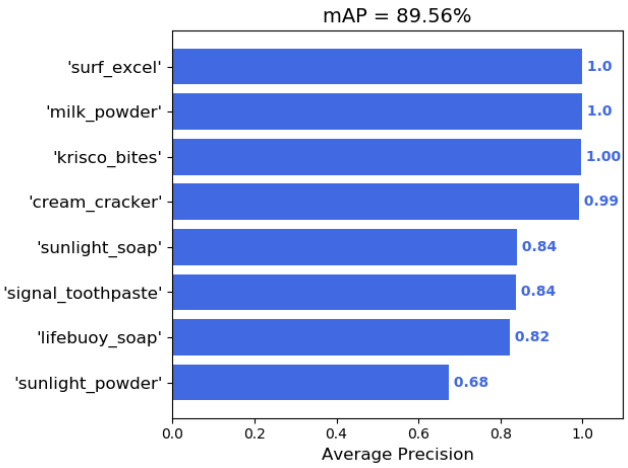


Figure 9 Average precision graph for Model 3 (YOLOv3 with a Darknet-53 + Semi-supervised Pseudo Labeling).

V. Conclusions and Future Work

Recent developments in transfer learning with object detection models have opened various avenues for the application of Computer Vision to real-world tasks. The main contribution of our work is the introduction of a fine-tuned object detection model that can accurately detect multiclass objects with obscure and overlapping boundaries. Additionally, we propose a pseudo labeling technique that can be applied to various domains to extend unlabeled datasets efficiently while minimizing mislabeled samples. We show the best model for overlapping object detection by comparing the performance of YOLOv3 and Faster R-CNN multiclass object detection methods and the effectiveness of semi-supervised learning approach to enhance the accuracy of the model.

This study focuses on bounding box annotations for object detections. Bounding box annotation can be applied to almost any conceivable objects. However, instance and semantic segmentations take object detection a step further. Rather than drawing a bounding box around the objects, instance segmentation annotation goes to pixel-level annotation [34]. Semantic segmentation also assigns pixel-level annotations [35]. Instance segmentation requires the identification and segmentation of individual instances in an image and semantic segmentation requires all the pixels in the image based on their class label. These segmentation methods go further with panoptic segmentation which is a combination of instance and semantic segmentation [36]. In the panoptic segmentation task, we need to classify all the pixels in the image as belonging to a class label, yet also identify what instance of that class they belong to as shown in Fig. 10. Mask R-CNN is an instance segmentation technique that locates each pixel of every object in the image instead of the bounding boxes [37]. We expect to improve the performance of overlapping object detection by using panoptic segmentation with Mask R-CNN technique to increase the object detection accuracy from pixel level for obscure and overlapping boundaries. Future work also includes augmentation of the data set with the pseudo labeling technique to increase the size of the dataset.

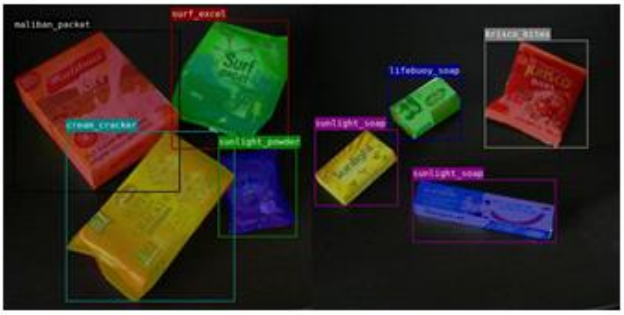


Figure 10 Panoptic segmented data.

REFERENCES

[1] C. Kamusoko, "Image classification," in Springer Geography, 2019.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014.

[3] C. J. Du and Q. Cheng, "Computer vision," in Food Engineering Series, 2014.

[4] M. Shah and R. Kapdi, "Object detection using deep neural networks," in Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017, 2017.

[5] C. Vision and I. P. Toolbox, "Edge and Corner Detection," Computer (Long. Beach. Calif)., 2005.

[6] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," IEEE Transactions on Neural Networks and Learning Systems. 2019.

[7] K. Lenc and A. Vedaldi, "R-CNN minus R," 2015.

[8] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015.

[9] E. Hanna and M. Cardillo, "Faster RCNN," Biol. Conserv., 2013.

[10] D. Yoo, S. Park, J. Y. Lee, A. S. Paek, and I. S. Kweon, "Attentionnet: Aggregating weak directions for accurate object detection," in Proceedings of the IEEE International Conference on Computer Vision, 2015.

[11] W. Liu et al., "SSD: Single shot multibox detector," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[13] K. Drid, M. Allaoui, and M. L. Kherfi, "Object detector combination for increasing accuracy and detecting more overlapping objects," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," Int. J. Comput. Vis., 2010.

[15] K. Sinha, "Semi-supervised learning," in Data Classification: Algorithms and Applications, 2014.

[16] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," Informatica (Ljubljana). 2007.

[17] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019.

[18] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," ICML 2013 Work. Challenges Represent. Learn., 2013.

[19] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "Imagenet," Adv. Neural Inf. Process. Syst. 25, 2012.

[20] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," IEEE Trans. Pattern Anal. Mach. Intell., 2017.

[21] W. Transfer, L. Now, T. L. Scenarios, and T. L. Methods, "Transfer Learning - Machine Learning's Next Frontier," PPT, 2017.

[22] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[23] Q. Zhang, Y. N. Wu, and S. C. Zhu, "Interpretable Convolutional Neural Networks," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[24] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[25] T. K. Trivedi, M. Glenn, K. Sporer, and G. Hern, "You Only Look Once," Ann. Emerg. Med., 2017.

[26] J. Redmon and A. Farhadi, "Yolov3," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2017.

[27] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in Proceedings of the IEEE International Conference on Computer Vision, 2019.

[28] Microsoft, "VoTT: Visual Object Tagging Tool," GitHub repository, 2018. .

[29] Y. Zhao, R. Han, and Y. Rao, "A new feature pyramid network for object detection," in Proceedings - 2019 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2019, 2019.

[30] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, 2019.

[31] J. Hui, "mAP (mean Average Precision) for Object Detection," Medium, 2018.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," J. Big Data, 2019.

[33] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," Constr. Approx., 2007.

[34] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path

Aggregation Network for Instance Segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[35] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-Time fusion network for rgb-d sementic segmentation incorporating unexpected obstacle detection," IEEE Robot. Autom. Lett., 2020.

[36] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., 2020.

Sentiment Classification of Social Media Data with Supervised Machine Learning Approaches: Common Framework, Challenges, and New Dimensions

Suresha N. Perera[1], Kasun Karunanayaka[2]
[1]Department of Computer Science, Open University of Sri Lanka, Nawala, Sri Lanka
[2]University of Colombo School of Computing, Reid Avenue, Colombo 7, Sri Lanka
Email: [1]waper@ou.ac.lk, [2]ktk@ucsc.cmb.ac.lk

***Abstract*—As an interdisciplinary research field, sentiment analysis is one of the momentous applications in Natural Language Processing, for quantifying the emotional value of text available in social media networks. The mission of this paper is to present a comprehensive overview of recent Machine Learning sentiment analysis classification techniques (Naive Bayes, Support Vector Machine, Rough-Fuzzy based classifiers etc.) to serve the scholars and researchers by emphasizing the methods used in current research. However, there is a minimal number of review papers discussed rough-fuzzy classifier involvement by researchers in sentiment analysis and there is a plethora of work that must be done with text mining. In addition to those, we propose a common framework for sentiment analysis in the context of social media based on previous works by providing the facility for users to enhance it with new concepts. Finally, discuss various research challenges and possible future directions in sentiment analysis.**

Keywords—Sentiment Analysis, Rough-Fuzzy Classifiers, Natural Language Processing, Machine Learning, Social Networking Sites, Introduction

I. INTRODUCTION

Decision-making behavior in an online community is predominant because many individuals now use community-based web services thus the opinions of others are readily available in online environments where people often rely on other individuals' decisions for social validation to make their own. For instance, individuals are inter-ested in others' opinions about political candidates before making a voting decision in a political election. In order to gather the public and consumer opinion organizations conduct surveys, opinion polls, or focus groups. The power of Social Media Websites (SMW) is rampant and growing a plethora of information and data convoluted with varying interests, opinions, and emotions with human generated baselines. The wide-spread use of the above media encourages positive and negative attitudes about people, organizations, places, events, and ideas. The growth of social media usage has led to an increasing accumulation of data and it is used for the prediction of the outcome as this platform becomes the most powerful communication media on the internet while hundreds of millions of messages are being posted every day.

SA, also called opinion mining, which is the field of computational study that analyzes people's opinions, attitudes and emotions toward an entity. It repre-sents a large problem space with many names and slightly different tasks, e.g., SA, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining, etc. There are three different levels of SA that has been investigated as:

Document Level Analysis - Classify whether a whole opinion document ex-presses a positive, negative or neutral sentiment [1], [2].

Sentence Level Analysis - Polarity is calculated for each

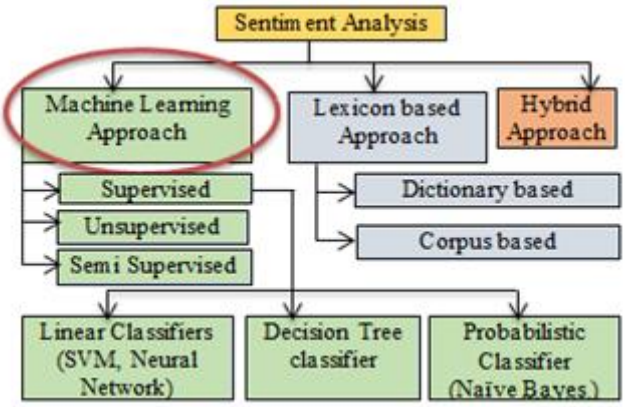


Fig 1 Three main approaches of the Sentiment Classification Method and it is further classified into subcategories. Only the ML branch is considered in this study.

while determining whether each sentence expressed a positive, negative, or neutral opinion.

Entity/Aspect Level Analysis - This discovers sentiments on entities and/or their aspects.

The three main approaches addressing SA [3] with social media are Lexicon based approach, ML based approach and Hybrid based approach [4] as shown in Fig. 1.

Lexicon based approach utilizes a sentiment lexicon to describe the polarity as positive, negative or neutral of a textual content and need the involvement of a human being at the analysis phase. This approach can be further divided into two categories: Dictionary based approach and Corpus based approach. ML approach can be divided into supervised, unsupervised and semi-supervised methods; it requires a large data set to be effective which is the main drawback. The classifier is trained on labeled data similar to test data with supervised learning. With unsupervised classifications, it is assigned labels based only on internal differences between the data points. ML provides more accuracy than lexicon based approach. Many researchers focused on Naive Bayes (NB) classification and Support Vector Machine (SVM) under the ML approach. Hybrid approach is the amalgamation of both ML and lexicon-based methods where many scholars are focusing nowadays. Supervised ML techniques have shown relatively better performance than the unsupervised lexicon based methods

[5]. Most of the algorithms used in ML approach belong to supervised classification and provide higher accuracy and performance [6], [7].

This paper presents a comprehensive literature review on different text mining and SA approaches with ML techniques in distinct areas by identifying future directions on research. A common framework will be discussed in this research paper for the use of researches to start SA with social media data. There are many challenges with SA applications which direct for new opportunities. This becomes more challenging as this is to determine the emotional state of a person with data mining concepts. Finally, the difficulties and challenges of text SA are discussed. SA on SMW data has become increasingly popular among the academics and they have undertaken a diverse range of related research.

This paper is articulated as follows: Section II presents related work. Section III describes a classification framework with supervised ML techniques. Section IV discusses challenges arising of ML techniques for SA. Finally, Section V leads to the conclusion.

II. LITERATURE REVIEW

There exists substantial research on SA and most active research on the area came with the explosion of user-generated content in social media, discussion forums, blogs and reviews. Since most studies use or depend on ML approaches, the amount of user generated content provided unlimited data for training. This literature review provides an overview of the different approaches that can be applied for SA as well as brief explanations of algorithms used by researchers.

To predict heart disease, Srinivas et al. [8] have exploited a rough-fuzzy classifier. It combines rough set with fuzzy set theory. The rules were generated using rough set theory and prediction done using a fuzzy classifier. The rule generation and relevant attribute identification are automated with rough sets and rules are automatically applied to fuzzy classifier for prediction of heart disease. The above classifier can be extended by using the associative analysis to find the relevant attribute and incorporating statistical measures to strengthen the computation. Keerthika et al [9] construct a Temporal Fuzzy Rule Based Classifier (TFRBC) which is built by the generalization of Fuzzy Rough Sets and uses fuzzy rough set. Further they have used temporal logic for mining temporal patterns in medical databases. After comparing the accuracies of classifiers such as FNN (Fuzzy Neural Network) and TFRBC it can be concluded that the accuracy of TFRBC is more with 88% when compared with the FNN which is 74%. As a future direction, the rule based classifier can be improved with effective decisions by taking upper approximations while reducing the number of rule sets when compared with the lower approximations.

Srividya and Sowjanya [10] discussed a methodology to analyse collected reviews over a period of time in Facebook using NB classifier into relevant and non-relevant to determine public opinion on the popularity of android based and identified the most preferable versions of OS for android phones and found Android KitKat is more preferable than others.

Election forecasting has a political side which helps in forecasting the results. In order to forecast an election there are opinion polls as well as used and many scientifically proven statistical models [11]. However sometimes polls also fail in predicting the results of the election even in the developed countries. According to [12], it listed several failed polls results such as in the 1992 British General Elections, French presidential etc.

Electoral analysis of twitter data is straightforward and optimistic even though it is a challenge for the research community in today's world. The outcome of the election with twitter data was analyzed by Salunkhe and Deshmukh [13] with the US and Gujarat Rajya Sabha election. The data collection was based on two methods: keyword match with tweets and using twitter API the collected data were pre-proceed with lower case conversion, punctuation and number removal, stemming and striping white spaces. NB used on the above training data set including emotions for the sentimental analysis. The dictionary based approach with eleven lexicons used for the classification of identified variables. The case study analysis concludes that SMW like twitter can be used in elections to predict the outcome well in advance. Further, this helps to explore the sentiment or views of citizens who have the voting power. Moreover, it can be successfully influenced by voters.

Wongn et al. [14] used twitter streaming API to collect tweets which contain the identified keyword relevant to the events identified before the election and used lexicon-based SA package, to extract the sentiment of tweets as a ternary (positive, negative, neutral) classification. They used the consistency relationship between tweeting and retweeting behavior. It is evident that the proposed method operates at much faster than existing methods. However, it does not require the explicit knowledge of the twitter network. As future directions they proposed to get the use of retweet matrix and retweet average scores when developing new models and algorithms.

Vadivukarassi et al. [15] proposed a model to analyse twitter data and preprocessed using Natural Language Toolkit (NTLK) techniques. The word scores of the features are tested based on Chi-square method and key words were scored sentiments during the analysis of data. NB classifier is used for training and testing the features and also evaluated the sentimental polarity and generated summarized report about the opinion from twitter.

Prabhu et al. [16] proposed a methodology to distribute political party's tickets during the election with twitter data. Upon receiving all the necessary data related to a candidate, NB Algorithm used to predict the most deserving candidate. Their final conclusion was that this classifier is suitable for any type of election in India and elsewhere.

Pak and Paroubek [17] proposed a classifier with collected twitter corpus to conduct linguistic analysis. This classifier tested only for the English language and the proposed technique can be used with any other language as future work. For the analysis of the corpus, they used a plot of word frequencies with Zipf's law inorder to understand how terms are distributed across collected corpus and TreeTagger [18] which is a tool for annotating text with part-of-speech and lemma information. Under the training of the

classifier they followed the method of filtering, tokenization, removal of stop words and constructing n-grams from twitter posts. As the classifier NB was selected and it outperformed with best results. The accuracy was established by discarding common n-grams with two strategies: computing the entropy of a probability distribution of the appearance of an n-gram in different datasets [19] and introducing a new term "salience" calculated for each n-gram and used multinomial NB.

Rao et al. [20] identified the importance of identifying the general sentiment polarity of a news article before publishing. They use a ML approach on twitter data with different features like unigram, bigram and hybrid. It shows that the hybrid feature with SVM classifier gives the best results for prediction of sentiment of twitter data. This Classifier can be further developed to make automatic sentiment classifiers for more than one language.

Wei and Gulla [21] present an analysis technique based on a tree of feelings of ontological features. The product's attributes labeling is handled with the novel HL-SOT approach. Hierarchical Learning (HL) process used for analyzing their associated sentiments in product reviews. Further they used a defined Sentiment Ontology Tree (SOT) with the above process. The HL-flat approach ignores the hierarchical relationships among labels when training each classifier and this is a "flat" version of HL-SOT. The H-RLS algorithm only uses identical threshold values for each classifier in the classification process where HL-SOT enables the threshold values to be learned separately for each classifier in the training process. The research found that the HL-SOT approach outperforms two baselines: the HL-flat and the H-RLS approach. The classification approach used is based on hierarchical classification algorithms.

Pang and Lee [2], [22] conducted a study to compare the performance of NB, Maximum Entropy and SVM for SA based on different features. For instance they considered only unigrams, bigrams, combination of both, incorporating parts of speech and position information, taking only adjectives etc. It is observed from the results that the feature presence is more important than feature frequency. Further, the accuracy falls when using bigrams. Moreover the accuracy improved when considering all the frequently occurring words from all parts of speech. Furthermore, the accuracy improved with not only with

adjectives but also with including position information. It was evident that NB performs better than SVM when the feature space is small, but SVM's perform better when feature space is increased.

A study has been conducted [23] on Pang Corpus which is a movie review database and opinions collected from the website Epinions.com named as Taboada Corpus to train a sentiment classifier with SVM together with n-grams and different weighting schemes: Term Frequency Inverse Document Frequency (TFIDF), Binary Occurrence (BO) and Term Occurrence (TO). Further, it uses chi-square weight features to select informative features. It is evident that the chi-square feature selection improves the accuracy of the classification. Further, it shows that the unigrams outperform the other n-gram models for both datasets. Table 1 depcted the summary of some of the researches conducted by scholars with their used algorithms, outcomes, advantages, dis-advantages and futre works.

Table 1. Sentiment analysis of social media data with various supervised learning techniques with their used algorithms, outcomes, advantages, dis-advantages, and future works

Study	**Dataset**	**classifier/** **Algorithm**	**Outcomes/Advantages/** **Disadvantages/Future works**
[8]	Cleveland, Hungarian and Switzerland datasets for heart disease prediction	Rough-fuzzy classifier	- Outperformed the previous approaches **Future work:** - Can be extended by including the associative analysis to find the relevant attribute

			- Rule strength computation can be extended by including statistical measure
[9]	Diabetic dataset to mine temporal patterns in medical databases	Fuzzy Temporal Rule Based Classifier	**Advantages:** - The minimum number of human involvement - Classifier is simple **Future work** - Build the rule based classifier with upper approximations with effective decisions to reduce the number of rule sets - Use less number of rule sets in order to build an efficient rule based classifier
[13]	Tweets in US and Gujarat Rajya Sabha election	NB with dictionary based approach	- Make prediction of future outcome of the election - Extract the sentiment or views of people - Sentiment analysis to classify their sentiment.

[15]	Tweets in 2012 U.S. presidential election campaign	Chi-Square test and NB classifier	- When the number of features increases, the accuracy of the selected features also increases. - Easy to generate summary report about the opinion from Twitter
[17]	Real Twitter posts	Multinomial NB classifier that uses N-gram and POS-tags as features	**Advantages:** - Improve the performance of the system by increasing the sample size - Best performance is achieved when using bigrams - When filtering out the common n-grams: salience provides a better accuracy than the entropy - Attaching negation words when forming n-grams increase performance **Future work:**

			use a multilingual corpus of twitter data and compare the characteristics of the corpus across different languages.
[20]	Extract twitter data using twitter API with analysis of news data	NB, SVM and maximum entropy using unigram, bigram and hybrid feature	- SVM using hybrid features outperforms the selection feature with accuracy of 84%. **Future work:** - Develop automatic sentiment classifiers for more than one language starting from the Hindi language.
[21]	Sentiment analysis on reviews of one product on customer reviews on digital cameras that are collected from a customer review website	A Hierarchical Learning (HL) process with a defined Sentiment Ontology Tree (SOT)	- HL-SOT approach outperforms two baselines: the HL-flat and the H-RLS approach **Future work:** - HLSOT approach can be easily generalized to labeling a mix of reviews of more than one products Advantages:

			- Separately learning threshold values for each classifier improve the classification accuracy - Knowledge of hierarchical relationships of labels improve the approach's performance - Product's attributes could be a useful knowledge for mining product review texts - lassification performance will be affected by variances of the generated SOTs an automatic method to learn a product's attributes and the structure of SOT from existing product review texts improve the efficiency

[23]	Pang Corpus and Taboada Corpus	SVM, N-grams and different weighting scheme, Chi-Square weight features to select informative features	- ChiSquare feature selection provide significant improvement on classification accuracy - Unigrams outperform other n-grams models - Binary Occurences (BO) and TFIDF weighting scheme plays a crucial role in extracting the most classical features in the data set - Accuracy is higher when the number of features selected is fewer - Performance varies with the domains and corpus size
[27]	Political tweets in presidential elections in Egypt 2012 for Arabic text classification	SVM and NB, with TF-IDF	- Obtained higher accuracy and performance with NB than SVM **Future work-** - Use Khoja stemmer and compare the result

			- Compare the result between unigram, bigram and trigram.
[30]	Twitter data	Competitive model that compares Linear (SVM) and probabilistic approach (Logistic Regression and NB)	- Observed that the performance and the accuracy is high with SVM **Future works** - Need more efficient machine learning, deep learning algorithm for better classifiers. - Deal with spam post/tweets.. - Use better mining techniques to deal with natural language processing more efficiently. **Findings:** **Advantages of SVM:** - Effective in high dimensional spaces.

			- Effective in cases where number of dimensions is greater than the number of samples. - Memory efficient. - Versatile **Disadvantages of SVM:** - Poor performances when the number of features is much greater than the number of samples. - Does not directly provide probability estimates and use an expensive five-fold cross-validation. **Advantages of NB:** - Easy and fast to predict class of test data set. - Performs well in multi class prediction. - Need less training data. - Performs well with categorical input variables compared to numerical variable(s).

			Disadvantages of NB: - Zero Frequency problem. Can use smoothing technique like Laplace estimation to solve this - The assumption of independent predictors.
[31]	Movie reviews from www.imdb.com and hotel reviews from OpinRank Review Dataset k+Review+Dataset)	K-Nearest Neighbour and NB algorithms	- Observed that the NB approach giving above 80% accuracies and outperforming than the k-NN approach. - Accuracies are lower for hotel reviews, in that both the classifiers yielded similar results. - NB classifier can be used successfully to analyse movie reviews **Future work -** - Compre with efficient sentiment analyser like random forest, Support vector Machine etc.

			- Implement a new algorithm utilizing the benefits of the both algorithms
[32]	Drug review analysis is collected by scraping from the raw HTML files using the Beautiful Soup Library in Python from Druglib.com	Use of fuzzy rough feature selection	- Fuzzy-rough feature selection significantly reduce the complexity of feature space - Reduce the classification run-time overheads while maintaining classification accuracy - Proposed framework bring forward more significant cost-efficiency savings to real-world healthcare analysis on large scale data Future work- - Exploit search strategies to increase the overall performance

			- Investigate the use of alternative approaches for learning classifiers which work better while dealing with the uncertainty inherent in natural language processing
[41]	The hashtagged (HASH) and emoticon (EMOT) as training datasets.	AdaBoost classifier, Unigrams, bigrams, lexicon, POS features, and micro-blogging features	- An F-measure of 0.68 was achieved for HASH. In addition, an F-measure of 0.65 was obtained by AdaBoost for HASH and EMOT datasets with a combination of n-grams, lexicons and microblogging features - part-of-speech features may not be useful for sentiment analysis in the microblogging domain

			- An existing sentiment lexicon features were somewhat useful in conjunction with microblogging features. However, the microblogging features are the most useful. - hashtags are useful in collecting the data set.
[43]	Labeled movie dataset	NB and SVM algorithms Use CountVectorizer and TF-IDF	- SVM classifier outperforms every other classifier - Compare results of SVM with other supervised learning algorithms such as maximum entropy classifier, Stochastic gradient classifier, K nearest neighbor and others.

An optimized classifier proposed by Bhumika and Vaghelawith [24] evident that SVM outperformed existing systems. The study was conducted for the movie review, twitter and Gold dataset between Optimized SVM towards SVM and NB classifier. Modifying hyper parameter values of RBF kernel SVM gives better results compared to SVM and NB algorithms. The SVM was implemented with RBF kernel hyper parameter (C, γ) where those parameters are modified with different combinations of regularization Constant (Soft Margin) C , kernel hyperparameter γ (gamma). The proposed approach has found optimal value for hyper parameters obtaining more accuracy. In [25], a SA framework used to analyze the performance of SVM for textual polarity detection with pre-labeled two twitter and one IMDB reviews data sets adhering to four main steps as: data set, preprocessing, classification and results. For performance evaluation of SVM, three ratios of training data and test data are used: 70:30, 50:50 and 30:70. Research concluded that performance of SVM depends upon the dataset as well as on the ratio of Training and Test Data.

A recent research [1] has been conducted to analyze amazon product review dataset obtained from the UCI repository with the probabilistic classifier NB. The research has concluded that the NB classifier is a highly scalable, simple classifier technique which provides better level of accuracy and good results after classification. Further, it identified this technique can be applied to any kind of review dataset.

A case study conducted to carry qualitative analysis on SMW data related to political leaders to identify different sentiments [26]. The implemented multi-label classification algorithm is capable of classifying polarity. The results show that Tuning Multinomial NB performs better than NB. The survey revealed social media data on political domain which overcomes the major drawback of both manual qualitative analysis and large scale computational analysis of user generated textual content. Another study of twitter data [27] which focuses on presidential elections in Egypt 2012 was conducted and revealed that NB scores the highest accuracy and the lowest error rate for Arabic text classification. It opens new research areas: compare the results with other classifiers, use Khoja stemmer and compare it with light stemmer and compare the results with bigram and trigram. Ringsquandl and Petkovic [28] conducted a study to analyze

the presidential candidates of the Republican Party in the USA and their campaign topics. It found out that the special considerations in retrieval and pre-processing are needed, NLTK's built-in pre-processing functionalities were not sufficient for informal text corpora. As future work they identified the need of learning other domain-specific opinion words like nouns and verbs. Kassraie et al. [29] conducted a research on predicting the US 2016 elections results and found that not all of the voters are twitter and google users, social media isn't always reliable, having active spammer robots, etc. In order to control this as a future plan, the user behavior was tracked over time for validating the consistency or trend of their opinion.

Raghuwanshi and Pawar [30] performed a comparative study on NB, SVM, and Logistic regression with crowd source information that compares linear and probabilistic approach. The results revealed that SVM turns out to be best among all and can work with linear or non-linear data. Processing and extracting exact emotions are two major areas to work in this field, need more efficient machine learning, deep learning algorithms for better classifiers, a lot of ways to deal with spam posts/tweets and use better mining techniques to deal with NLP more efficiently.

A web crawling framework to facilitate the quick discovery of sentimental contents of movie and hotel reviews conducted with two supervised machine learning algorithms: K-Nearest Neighbour (K-NN) and NB [31]. For movies review, NB gave far better results than K-NN but for hotel reviews these algorithms gave minor, almost similar accuracies. The researchers suggested testing the results with random forest, SVM etc. and trying to implement a new algorithm utilizing the benefits of the both algorithms.

One of the challenges with SA is the large feature space extracted. Many researchers focus towards this area and Fuzzy Rough Set based approaches provide substantial amounts of solutions to overcome them. Chena et al. [32] proposed an approach for drug reviews using fuzzy rough feature reduction. They used Random Forest algorithm with fuzzy-rough QuickReduct feature selection to obtain significantly reduced features and it improved the accuracy. Further they noticed that it does not consume additional run-time overheads and subsequently improve the performance. It is evident that Fuzzy Rough Feature Selection (FRFS) showed promising results and this is a nice start for the future researchers. They believed that this framework may bring tremendous benefits for the real world health sector with cost-efficiency savings to real-world healthcare analysis on large scale data with enormous data. As future development, the overall performance of the framework can be increased while focusing on alternative approaches for learning classifiers to ensure accuracy when dealing with NLP.

A fuzzy rough set-based feature selection algorithm has been used for hierarchical feature selection with sibling strategy and revealed that it is more efficient and more versatile [33]. Further it shows that the classifier resulted with higher performance with establishing the efficiency and effectiveness. It opens new research trends by combining fuzzy rough sets with hierarchical feature selection problems.

Many researchers now combine fuzzy rough set theory with other technologies to improve classifiers. Genetic Search Fuzzy Rough (GSFR) is one of the feature selection algorithm used by [34] using the evolutionary sequential genetic search technique with fuzzy rough set theory to early identification of cancer. This is an extended approach of Fuzzy-rough nearest neighbor (FRNN) classifier. This classifier outperforms with number of features, accuracy, and precision, recall, F-measure and computation time compared to other classifiers. This research opens new doors to hybridize fuzzy rough sets with both Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) in order to improve further.

A new approach based on FRNN proposed by R. Jensen and Cornelis [35] tested over nine data sets. FRNN-FRS uses the traditional operations, a t-norm and an implicator. FRNN-VQRS is the fuzzy quantifier-based approach used. It resulted that, FRNN outperforms both FRNN-O, as well as the traditional fuzzy nearest neighbour (FNN) algorithm. This research opens new investigations by providing explanations of the impact of the choice towards the fuzzy relations, connectives and quantifiers. The accuracy of the classifier upon the feature selection preprocessing is another new research area arising from this.

The above literature review indicates that a sufficient amount of studies have been conducted by various authors during the last decade. It explored that the SA in previous work was to find out whether the expressed opinion in the document or sentence is positive, negative or neutral. Many researchers have studied different techniques for sentiment analysis like NB classifier, SVM algorithm, Rough-Fuzzy classifiers etc. for the SA. NB performed well with most of the cases and easy to implement. SVM obtained favorable results with prediction accuracy while bringing fast evaluation of the learned target function. However, it consumes much computational time, is expensive in memory and requires long training time [36]. The text pre-processing plays a vital role in terms of the accuracy and the performance of the classifier. Interestingly, it has been found that the n-grams used as features in classification improve the result, in some cases the unigrams perform well. Furthermore, the researchers conduct their studies in different domains (product reviews, political sentiment, and news). Decent amount of related prior work has been done in SA. They have contributed to this domain a lot and there are still some

gaps to be filled as mentioned in the literature review. Hence sentiment analysis has become a popular field for research work. It is very useful for academic as well as for business purposes.

III. TEXT CLASSIFICATION FRAMEWORK WITH SUPERVISED ML TECHNIQUES

Through the knowledge observed with the literature review, an abstraction model of a generic framework for SA in the context of social media depicted in Fig. 2.

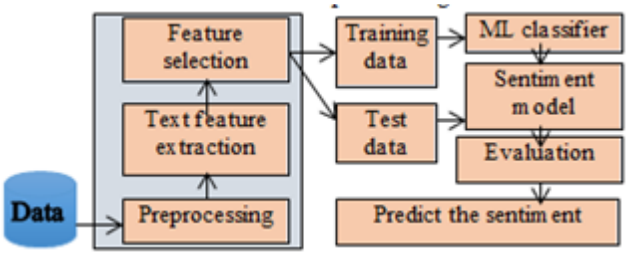


Fig 2 A common Framework for SA - This is a general model for the SA consisting of a number of phases. Both training and testing data should be pre-proceed. Then the feature extraction and selection should be done. The ML classifier developed and trained with

Data preparation plays a vital role since the real data improves the accuracy and the performance. SNW contains data such as comments, tweets which are unstructured representing the opinion of people. Thus, the data preprocessing needed to be done by removing unimportant or disturbing elements such as URLs and hashtags in tweets. Furthermore, remove punctuations, extra blank spaces and vowels repeated in sequence at least three times, convert emoticons into tags, convert text to lower case [37] are necessary follow up states. Stop word removal helps faster processing as well as in dimension reduction in terms of space requirement. Next, commonly used normalization procedures: stemming and lemmatization are to be done [38]. At the final stage the tokenization [39] has to be done in order to split a string into a list of tokens.

Text features are extracted from the above pre-

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (4)$$

processed data. Many researchers use various techniques such as Term Frequency-Inverse Document Frequency (TFIDF) , Bag Of Words (BOW), word embedding, word count, noun count etc [40]. At the end of this step the researchers found that the amount of features extracted is too large to be used directly. Hence, the feature reduction is important at this stage. Chi-square score is a feature reduction technique used by many as the attribute evaluation metric in order to obtain the high information features [41], [42]. Now the Corpus will be split into two data sets, training and test. The training data set will be used to fit the model and the predictions will be performed on the test data set. The data set is now passed into a classification algorithm such as NB, SVM etc. After training the model, the evaluations have to be done for the classifier.

A confusion matrix [43], also known as an error matrix is a special kind of contingency table with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions that allows visualization of the performance of an algorithm, typically with supervised learning as shown in Table 2 where: P = positive; N = Negative; TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative.

Table 2 The confusion matrix in abstract terms

		Actual Class	
		P	N
Predicted class	P	TP	FP
	N	FN	TN

From classification point of view TP, FP, TN and FP are used to compare labels of classes [44]. Based on the data of the confusion matrix, following measures are suitable for evaluating performance of classifiers.

Precision measures the exactness.

$$Precision = \frac{TP}{TP + FP}$$

Recall measures the completeness/sensitivity.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

F-measure combines both precision and recall. The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

Accuracy is the portion of all true predicted instances against all predicted instances.

Almost all classifiers developed by researchers were evaluated empirically by using above indexes.

Now the classifier is ready for the prediction. The above proposed framework could serve a basis for future works as an extensible and complete guideline for SA.

IV. CHALLENGES AND FUTURE WORK

There are some weaknesses and limitations within the domain of the sentiment classification identified as obstacles that affect the efficiency and effectiveness of the classifier.

The detection of spam and fake reviews - This is always difficult with human centric corpus.

Duplicates of the same tweet presents.

Word Sense Disambiguation (WSD) - Since this is human centric and opinion based, a word that is considered to be positive in one situation may be considered negative in another situation. Therefore, WSD is a basic and on-going issue that occurs with text mining [45]. Different people express their opinions in different ways. There might be a small difference between the texts, however the meaning can be drastically different from the opinion.

Identifying the entity - This is one of the challenges in opinion mining. A text may have multiple entities associated with it. In the same text it may represent both negative and positive polarity.

The negation problem - This is a problem which can direct to a completely wrong decision as it reverses the polarity of the word.

The intensity of the opinion - This is the degree of polarity which becomes problematic when finding the sentiment.

Sudden changes in the same tweet.

The domain - In some cases, the domain is different and it has hidden meaning for some of the words where context matters.

Demographics bias - This is associated with the comments/tweets and however it was neglected even when it is well known that social media is not a random sample of the population.

Humor and sarcasm - Some people use positive or intensified positive words to express their negative feelings towards an entity. This plays a major role and should take precautions for that.

Comparisons - It is difficult to explore the polarity when the tweets are to compare two entities.

There are many future directions associated with SN.

Expansion for multiple languages - Analyzing sentiment using the only English language (the most common lexicon source is WordNet) isn't worthy of every time since many people use their native language to communicate via SMW and this opens a new research area to create lexica, corpora and dictionaries resources in the context of other natural languages.

Feature selection and reduction plays a vital role in text mining. Feature space reduction approaches while ensuring the performance by removing redundancy be a future direction for research work [46].

The literature reveals that the Fuzzy Rough Sets based approaches effectively improved the accuracy and accomplished higher consistencies. Hence, a research gap arises to increase the overall performance together with above identified measures.

Find new approaches to guarantee accuracy since text mining always deals with NLP.

The recent finding shows the need of fuzzy rough sets based approaches and hierarchical feature selection problems to be addressed to improve the accuracy and the performance of the SA classifiers.

The hybrid concept is much prominent with SA and there are new research areas open for fuzzy rough sets with both Particle Swarm Optimization and Ant Colony Optimization in order to improve classifiers further.

- Experimentation of n-gram models with incorporating the contextual knowledge is rampant and another direction is to work with real live streaming text data.

V. CONCLUSION

It has been observed that the performance of the classifier depends on the algorithm, and various feature selection schemes. Thus no classifier alone provides complete accuracy and performance whereas need to combine additional features. The use of N-gram feature increases the accuracy of the classifier with improved prediction results as with increased sentiment length. Machine learning approaches surrender best results compared to other approaches in SA. There is a dependency with the domain where a trained classifier on a particular domain does not work accurately with another domain. There are many new research areas open in the field of rough fuzzy SA classification techniques as discussed in this paper such as: increase the accuracy by improving the overall performance, hierarchical feature selection, hybridize of fuzzy rough sets with both PSO and ACO. A common guideline in a structured way is necessary since many researchers actively work in this field and the proposed framework is fulfilling that. As a future work, this framework can be enhanced with new findings by providing the ability for users to plug new components where necessary. The authors wish to test this framework with a case study in the political domain to predict the sentiment towards the candidates in an election by using SMW.

REFERENCES

[1] Turney, P.D.: Thumbs up or thumbs down?, in Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL-02, 40th Annual Meeting of the Association for Computational Linguistics , Philadelphia, pp. 417-424, July 2002

[2] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 79-86, July 2002, doi: 10.3115/1118693.1118704

[3] Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. In: Ain Shams Engineering Journal, vol 5, no. 4, December 2014

[4] Payne, J.G.: The Bradley effect: Mediated reality of race and politics in the 2008 US presi-dentialelection. In: American Behavioral Scientist, vol. 54, no. 4, pp. 417–435, 2010

[5] Ceron, A., Curini, L., Iacus S.M., Porro, G.: Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. In: New Media and Society, vol. 16, no. 2, pp. 340–358, 2014

[6] Meghashree, K., Radhika, S., Shilpashree, A., Dinni, S., Monika, P.: Survey Paper on Algorithms used for Sentiment Analysis. In: International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 8, no. V, May 2020

[7] Devika, M.D., Sunitha C., Ganesha, A.: Sentiment Analysis:A Comparative Study On Different Approaches. In: Fourth International Conference on Recent Trends in

Computer Science & Engineering, Procedia Computer Science 87 (2016), pp. 44 – 49, 2016, doi: 10.1016/j.procs.2016.05.124

[8] Srinivas, K., Rao, G. R., Govardhan, A.: Rough-Fuzzy Classifier: A System to Predict the Heart Disease by Blending Two Different Set Theories. In: Arabian Journal for Science and Engineering, vol. 39, no. 4, pp. 2857–2868, 9 February 2014, doi: 10.1007/s13369-013-0934-1

[9] Keerthika, U., Sethukkarasi, R., Kannan, A.: A ROUGH SET BASED FUZZY INFERENCE SYSTEM FOR MINING TEMPORAL MEDICAL DATABASES. In: International Journal on Soft Computing (IJSC), vol.3, no.3, August 2012 doi: 10.5121/ijsc.2012.3304

[10] Srividya, K., Sowjanya, A.M.: Sentiment analysis of facebook data using naïve Bayes classifier. In: International Journal of Computer Science and Information Security (IJCSIS), vol. 15, no. 1, January 2017

[11] Beck, M.S.L.: Election forecasting: principles and practice. In: The British Journal of Politics and International Relations, vol. 7, no. 2, pp. 145-164, 2005

[12] Fumagalli, L., Sala, E.: The total survey error paradigm and pre-election polls: The case of the 2006 Italian general elections. In: ISER Working Paper Series, 2011

[13] Salunkhe, P., Deshmukh, S.: Twitter Based Election Prediction and Analysis. In:International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 10, October 2017

[14] Wong, F.M.F., Tan, C.W., Sen, S., Chiang, M.: Quantifying Political Leaning from Tweets, Retweets, and Retweeters. In: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, 2013

[15] Vadivukarassi, M., Puviarasan, N., Aruna, P.: Sentimental Analysis of Tweets Using Naive Bayes Algorithm. In: World Applied Sciences Journal,India, vol. 35, no. 1, pp. 54-59, 2017, doi:10.5829/idosi.wasj.2017.54.59

[16] Prabhu, B. P. A., Ashwini, B. P., Khan, T.A., Das, A.: Predicting Election Result with Sentimental Analysis Using Twitter Data for Candidate Selection. In: Innovations in Computer Science and Engineering, pp. 49-55, 19 June 2019 doi: 10.1007/978-981-13-7082-3_7

[17] Pak, A., Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, European Language Resources Association 2010, pp. 17-23 May 2010, ISBN 2-9517408-6-7, 1994

[18] Schmid, H.: Probabilistic part-of-speech Tagging Using Decision Tree. In: International Conference on New Methods in Language Processing, Manchester, UK

[19] Shannon, C.E., Weaver, W.: A Mathematical Theory of Communication. In: The University of Illinois Press, Champaign, IL, USA. 1963

[20] Rao, C.S., Prasad, G.S., Rao, V.V.: Prediction and Analysis of Sentiments on Twitter Data using Machine Learning Approach. In: International Journal of Computer Science and Information Security (IJCSIS), vol. 16, no. 8, August 2018

[21] Wei, W., Gulla, J.: Sentiment Learning on Product Reviews via Sentiment Ontology Tree. In: 48th Annual Meeting of the Association for Computational Linguistics. [online] Sweden: Association for Computational Linguistics, pp. 404–413, 2019

[22] Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: Proceedings of the ACL, 2004

[23] Zainuddin, N., Selamat, A.: Sentiment Analysis Using Support Vector Machine. In: International Conference on Computer, Communication, and Control Technology (I4CT 2014), September 2 - 4, 2014 - Langkawi, Kedah, Malaysia

[24] Bhumika, M.J., Vimalkumar, B.V.: Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis. In: International Journal of Computer Applications (0975 – 8887), vol. 146, no.13, July 2016,

[25] Ahmad, M., Aftab, S.: Analyzing the Performance of SVM for Polarity Detection with Different Datasets. In: Int. J. Mod. Educ. Comput. Sci., vol. 9, no. 10, pp. 29–36, 2017.

[26] Tarlekar, A., Kodmelwar, M.K.: Sentiment Analysis of Twitter Data from Political Domain Using Machine Learning Techniques. In: International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, no. 6, 2015, doi: 10.15680/ijircce.2015.0306084

[27] Elghazaly, T., Mahmoud, A., Hefny, H.A.: Political Sentiment Analysis Using Twitter Data. In: ICC '16: Proceedings of the International Conference on Internet of things and Cloud Computing, 2016, doi: 10.1145/2896387.2896396

[28] Ringsquandl, M., Petkovic, D.: Analyzing Political Sentiment on Twitter. In: 2013 AAAI Spring Symposium, 2013

[29] Kassraie, P., Modirshanechi, A., Aghajan, H.K., Election Vote Share Prediction using a Sentiment-based Fusion of Twitter Data with Google Trends and Online Polls. In: Proceedings of the 6th International Conference on Data Science, Technology and Applications, pp: 363-370, 2017

[30] Raghuwanshi, A.S., Pawar, S.K.: Polarity Classification of Twitter Data using Sentiment Analysis. In: International Journal on Recent and Innovation Trends in Computing and Communication, vol. 5, no. 6, pp: 434 – 439, 2015

[31] Dey, L., Chakraborty, S., Biswas, A., Bose, B., Tiwari, S.: Sentiment Analysis of Review Datasets Using Naïve, Bayes‘ and K-NN Classifier. In: International Journal of Information Engineering and Electronic Business, 2016, doi: 10.5815/ijieeb.2016.04.07

[32] Chen,, Su, T. P., Shang, C., Hill, R., Zhang, H., Shen, Q.: Sentiment Classification of Drug Reviews Using Fuzzy-rough Feature Selection. In: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2019. Available: 10.1109/fuzz-ieee.2019.8858916 [Accessed 11 May 2020].

[33] Zhao, H. , Wang, P., Hu Q., Zhu, P.: Fuzzy Rough Set Based Feature Selection for Large-Scale Hierarchical Classification. In: IEEE Transactions on Fuzzy Systems, vol. 27, no. 10, pp. 1891-1903, 2019. Available: 10.1109/tfuzz.2019.2892349 [Accessed 11 May 2020].

[34] Meenachi, L., Ramakrishnan, S.: Evolutionary sequential genetic search technique-based cancer classification using fuzzy rough nearest neighbour classifier. In: Healthcare Technology Letters, vol. 5, no. 4, pp. 130-135, 2018. Available: 10.1049/htl.2018.5041 [Accessed 11 May 2020]

[35] Jensen, R., Cornelis, C.: A New Approach to Fuzzy-Rough Nearest Neighbour Classification. In: Chan CC., Grzymala-Busse J.W., Ziarko W.P. (eds) Rough Sets and Current Trends in Computing. RSCTC 2008. Lecture Notes in Computer Science, vol 5306. Springer, Berlin, Heidelberg, 2008

[36] Patil, P., Yalagi, P.: Sentiment Analysis Levels and Techniques: A Survey. In: International Journal of Innovations in Engineering and Technology (IJIET), vol. 6, no 4, April 2016

[37] Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F., Manicardi, S. : A Comparison between Preprocessing Techniques for Sentiment Analysis in

Twitter. In: 2nd International Workshop on Knowledge Discovery on the Web-KDWEB 2016, 2016, Cagliari, Italy

[38] Batanovic, V., Nikolic, B.: Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization and Word Embeddings. In: Telfor Journal, Vol. 9, No. 2, 2017

[39] Kusrini, Mashuri, M.: Sentiment Analysis In Twitter Using Lexicon Based and Polarity Multiplication. In: 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), pp. 365-368, Yogyakarta, Indonesia, 2019 doi: 10.1109/ICAIIT.2019.8834477.

[40] Ahujaa, R., Chuga, A., Kohlia, S., Guptaa, S., Ahuja, P.: International Conference on Pervasive Computing Advances and ApplicaThe Impact of Features Extraction on the Sentiment Analysis. In: International Conference on Pervasive Computing Advances and Applications – PerCAA 2019, 2019, doi: 10.1016/j.procs.2019.05.008

[41] Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg!. In: Icwsm, vol. 11, pp. 538–541, 2011.

[42] Yang, A., Zhang, J., Pan, L., Xiang, Y.: Enhanced Twitter Sentiment Analysis by UsingFeature Selection and Combination. In: International Symposium on Security and Privacy in Social Networks and Big Data, 2015, doi: 10.1109/SocialSec2015.9

[43] Tripathy, A., Rath, S.K.: Classification of Sentiment of Reviews using Supervised Machine Learning Techniques. In: International Journal of Rough Sets and Data Analysis, vol. 4, no. 1, January-March 2017

[44] Mouthami, K., Devi, K.N., Bhaskaran, V.M.: Sentiment analysis and classification based on textual reviews. In: Proceedings of theInternational Conference on Information Communication and Embedded Systems (ICICES), pp. 271-276, 2013, doi:10.1109/ICICES.2013.6508366

[45] Wang, Y., Wang, M., Fujita, H.: Word Sense Disambiguation: A comprehensive knowledge exploitation framework. In: Knowledge-Based Systems (2019) 105030

[46] Asghar, M.Z., Khan, A., Ahmad, S., Kundi, F.M.: A Review of Feature Extraction in Sentiment Analysis. In: Journal of Basic and Applied Scientific Research, February 18 2014

Decision Supportive Index to Rank the Countries According to the Transmission Risk of COVID-19: A Fuzzy Mathematical Approach

AWSP Karunarathne [1] , ITS Piyatilake [2,]
[1] Department of Computational Mathematics, Faculty of Information Technology, University of Moratuwa, Sri Lanka
email: [1] sachinikarunarathne94@gmail.com

***Abstract*—COVID-19 has emerged as a global pandemic reporting more than 33.5 million of cases around the world. Many researchers are still investigating for the facts which has caused to the outbreak of this disease. Currently, researchers have identified some risk factors which cause to the spread of COVID-19 more severely. Some of them are, population density, gross domestic product per capita, available hospital beds, handwashing facilities in a country, life expectancy and human development index etc. The objective of this research is to rank selected countries according to the risk of the spread of COVID-19 considering the combined effect of these factors. Due to the uncertainty and the impreciseness of the the factors a fuzzy mathematics technique is applied. With the assistance of the extent analysis method the weights of the risk factors are evaluated. The ranks of the countries are obtained using TOPSIS. Among the selected countries United States, Sri Lanka, India, Philippines and Thailand were the top five countries which has highest risk of spreading COVID-19 cases.**

Keywords—COVID-19, Fuzzy, Transmission risk, Rank

I. INTRODUCTION

COVID-19 is an infectious disease and the first confirmed case has been diagnosed in December 2019 in Wuhan city, China [1]. This disease is caused by a novel virus named coronavirus. Due to the severity, it is identified as a deadly disease. According to the available records, 213 countries around the world have reported more than 33.5 million of COVID-19 cases and nearly 1.01 million of deaths [2]. The typical symptoms of this disease are fever, dry cough and tiredness [3]. When someone gets affected from this disease it takes 5 to 6 days to show the symptoms. Coronavirus could be spread to a healthy person by having a close contact with an infected person [4]. The most common direct coronavirus droplets transmission modes are coughs sneezes and speaks. Apart from the direct method, people can indirectly become infected by touching their nose, mouth, and eyes after contacting with the objects or surfaces that this virus lands. Still vaccine has not discovered, and antibiotics do not work against this virus. Therefore, currently there is no specific recognize treatment for this novel disease. When someone gets infected from the disease, he or she must have to stay in isolation until they recover from the disease. People have to follow cross-protection techniques which is called as pre-immunization methods in order to protect against this disease.

It can be seen that some of the countries in the world have controlled spreading the disease in the society by implementing control strategies such as impose a nationwide lockdowns, restrict travel from endemic countries, introduce self quarantine methods, conduct the session regarding hygiene methods and introduce work from home programs [16]. Meantime the factors such as low population density, high GDP per capita, health conditions of the citizens, available hospital facilities and the number of doctors allocated also helpful to control the transmission and deaths of the disease. However, the global numbers of patients with COVID-19 are still increasing, and no country seems to be spared from this dangerous situation. This pandemic situation will weaken the socioeconomic growth of the developing countries and its consequences are still being unknown. Therefore, it is significant to distinguish the countries observing the behavior of the transmission of COVID-19. It will also help health authorities and travelers to recognize which country is severely affecting and which country is going to be affected with this disease in future. Therefore, our aim is to develop a decision supportive index to rank the countries according to the transmission of COVID-19.

Since 2019 December, many researchers have started to investigate about the spread on this novel disease. Showmitra had identified the susceptibilities in risky areas of Bangladesh using multi-criteria evaluation techniques [24]. Iwendi et al. proposed a fine-tuned random forest model to predict the severity of COVID-19 patients [10]. For that purpose, they had used the patients geographical, travel, health, and demographic data. Ganya et al. had captured the future trend of coronavirus [11]. They used a 4+1 penta-grouped model for predictions and using a BAT model they had forecasted the most suitable return date for university students. Moreover, Arsalan et al. had developed a multi-factor weighted spatial analysis to identify how the countries have impacted by the coronavirus. Also, the mortality risk of the global was considered and they have resulted that 44/153 countries are experiencing 20% increase in mortality due to CIVID-19 [8]. Many more researchers have carried out numerous studies on COVID-19. Most of these models were developed to predict the number of patients and to identify the factors associated with this disease. Currently, the numbers of reported cases are used to rank the world countries according to the COVID-19 spread. So, it is worthwhile to carry out a study to measure the risk of spreading COVID-19 considering the combined effect of multi factors such as population density, GDP per capita, health conditions of the citizens, available hospital facilities and the number of doctors. This type of study also helps to investigate factors contribution towards the spread of COVID-19.

However, these sorts of factors are highly uncertain. We can't define the boundaries of these factors precisely. Therefore, aggregating these types of uncertain factors and ranking countries according to the results is a quite complicated task. However, Zadesh developed a mathematical tool in 1965 to handle these kind of uncertain environments and we called it as fuzzy mathematics [18]. Therefore, fuzzy mathematics concepts are used in this study to prioritize the multi factors and to rank the countries. Extent analysis method which was invented by Chang is

used to derive the weights of the selected risk factors. The technique for order performance by similarity to ideal solution (TOPSIS) which was invented by Hwang and Yoon is used to obtain the ranks of the countries [[19] – [23]] according to the disease transmission risk.

This manuscript is collated as follows: In Section 2, mathematical theories behind the model building process is presented. Selected variables, countries and ranking model is presented in Section 3. Section 4 is used to present the results and discussion. Finally, in Section 5 conclusions and future directions of this study are pointed out.

II. PRELIMINARIES

Mathematical theories namely fuzzy membership functions, operations on membership functions, extent analysis method and TOPSIS which are used in this research are presented in this section. These theories are obtained from the literature [18,24].

A. Definitions in Fuzzy Mathematics

Definition 1. Fuzzy Number

Let A be a fuzzy set in universe of discourse R then, m is a fuzzy number if and only if there exists a closed interval (which may be singleton) $[m,n] \neq \phi$

$$\mu_A(x) = \begin{cases} 1 & if\ x \in [m,n] \\ l(x) & if\ x \in (-\infty, m) \\ r(x) & if\ x \in (n, \infty) \end{cases} \quad (1)$$

Here l and r can be defined as follows:

(i) $l:(-\infty,m) \to [0,1]$ is monotonic increasing, continuous from the right such that,

$$l(x) = 0 \text{ for } x \text{ in } (-\infty, k_1),\ k_1 < m. \quad (2)$$

(ii) $r:(n,\infty) \to [0,1]$ is monotonic decreasing, continuous from the left such that

$$r(x) = 0 \text{ for } x \text{ in } (k_2, \infty),\ k_2 > n. \quad (3)$$

Definition 2. Triangular Fuzzy Number

A fuzzy number M on R is called a triangular fuzzy number if its membership function is in the below form.

$$\mu_M(x) = \begin{cases} \frac{x-l}{m-l}, & \text{if } x \in [l,m]; \\ \frac{u-x}{u-m}, & \text{if } x \in [m,u]; \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

Where,

l - smallest possible value of an event.

m - the most promising value of an event.

u - the largest possible value of an event.

Definition 3. Operations on Triangular Fuzzy Numbers

Let $P = (l_1, m_1, u_1)$ and $Q = (l_2, m_2, u_2)$ be two triangular fuzzy numbers. Then addition operation, multiplication operation and reciprocal value of a fuzzy number can be stipulated as follows:

$$P \oplus Q = (l_1 + l_2, m_1 + m_2, u_1 + u_2) \quad (5)$$

$$P \otimes Q = (l_1 l_2, m_1 m_2, u_1 u_2) \quad (6)$$

$$P^{-1} \approx \left(\frac{1}{u_1}, \frac{1}{m_1}, \frac{1}{l_1} \right) . \quad (7)$$

Definition 4. Pair-Wise Comparison Matrix

Let $\tilde{P}$ is a $n \times n$ decision matrix in a fuzzy environment and there are n number of factors to be evaluated. This matrix contains all pair-wise comparisons between factor i and j for all $i, j \in 1,2,3,...,n$. The importance of i^{th} factor with respect to j^{th} factors can be denoted as $\tilde{p}_{ij}$. Therefore, $\tilde{P}$ can be defined as follows:

$$\tilde{P} = \begin{pmatrix} (1,1,1) & \tilde{p}_{12} & ... & \tilde{p}_{1n} \\ \tilde{p}_{21} & (1,1,1) & ... & \tilde{p}_{2n} \\ : & : & : & : \\ \tilde{p}_{n1} & \tilde{p}_{n2} & ... & (1,1,1) \end{pmatrix} , \quad (8)$$

where $\tilde{p}_{ij}$ is a triangular fuzzy number and $\tilde{p}_{ij} = (1,1,1):\ \forall i = j,\ \tilde{p}_{ji} = \tilde{p}_{ij}^{-1}$

B. Extent Analysis Method

Let $O = \{o_1, o_2, ..., o_n\}$ be the object set and $G = \{g_1, g_2, ..., g_n\}$ be the goal set. Suppose each object is selected and extent analysis is performed for each goal. Therefore, now we have m extent analysis values for each object. Suppose these generated values are denoted as $E_{g_i}^1, E_{g_i}^2, ..., E_{g_i}^m$, $i = 1,2,...,n$. Here $E_{g_i}^j$ $(j = 1,2,...,m)$ extent values are taken as triangular fuzzy numbers. The following four steps have to be considered in extent analysis method in order to find the weights of the factors.

(1) Calculate fuzzy synthetic extent values: The synthetic extent value with respect to the i^{th} factor can be derived as follows:

$$S_i = \sum_{j=1}^{m} E_{g_i}^j \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} E_{g_i}^j \right]^{-1} . \quad (9)$$

To compute $\sum_{j=1}^{m} E_{g_i}^j$, we can use m extent analysis values of the decision matrix and

we can aggregate them using (5). In order to compute $\left[\sum_{i=1}^{n}\sum_{j=1}^{m} E_{g_i}^{j}\right]^{-1}$, we can use $E_{g_i}^{j}$ $(j=1,2,...,m)$ values and we can aggregate these values using (5).

(2) Compute and compare degree of possibilities values: Let $P_1=(l_1,m_1,u_1)$ and $P_2=(l_2,m_2,u_2)$ be two triangular fuzzy numbers such that $P_2 \geq P_1$. The value of degree of possibility of the two fuzzy numbers P_1 and P_2 can be defined as follows:

$$D(P_2 \geq P_1)=\begin{cases}1, & \text{if } m_2 \geq m_1;\\ 0, & \text{if } l_1 \geq u_2;\\ \dfrac{l_1-u_2}{(m_2-u_2)-(m_1-l_1)}, & \text{otherwise.}\end{cases} \quad (10)$$

According to the values of $D(P_1 \geq P_2)$ and $D(P_2 \geq P_1)$ we can compare the given two fuzzy numbers.

(3) Derive the weight vector: The degree of possibility values for a convex triangular fuzzy number to be greater than *k* convex triangular fuzzy number P_i, $(i=1,2,.....,k)$ can be defined as follows:

$$\begin{aligned} D(P \geq P_1, P_2, ..., P_k) &= D[(M \geq M_1) \text{ and } (M \geq M_2) \text{ and } .... (M \geq M_k)] \\ &= \min D(M \geq M_i),\ i=1,2,...,k \end{aligned} \quad (11)$$

Assuming $D'(A_i)=\min D(S_i \geq S_k)$ for $k=1,2,...,n: k \neq i$, we can derive the weight vector of the factors and it is given by,

$$W'=(D'(A_1), D'(A_2), ..., D'(A_n))^T \quad (12)$$

where, $A_i=(i=1,2,...,n)$ are *n* elements.

(4) Calculate the normalize weight: Normalizing the components values in (12) using $d(A_i)=\dfrac{D'(A_i)}{\sum_{i=1}^{n} D'(A_i)}$, we can obtain the normalize weight vector as follows:

$$W=(d(A_1), d(A_2), ..., d(A_n))^T. \quad (13)$$

Now the components of this normalize weight vector *W* are non-fuzzy numbers.

C. TOPSIS

This method is used to rank the countries according to the transmission risk of COVID-19. The following are the steps involved in TOPSIS method.

(1) Calculate normalize values of the given data: Given data values can be normalize as follows:

$$n_{ij}=\frac{x_{ij}}{\sqrt{\sum_{j=1}^{m} x_{ij}^2}}, \quad j=1,2,...,m \quad i=1,2,...,n \quad (14)$$

where x_{ij} is the value of the i^{th} factor for the j^{th} country.

(2) Find the weighted normalized values: Weighted normalized decision matrix is given by

$$W_N = [d(A_i) \times n_{ij}]_{n \times m}, \quad (15)$$

for $i=1, 2, ..., n$ and $j=1, 2, ..., m$.

(3) Compute the values of *PIS* and *NIS:* We can define the positive ideal solution (PIS) and negative ideal Solution (NIS) as follows:

$$\text{PIS } A^+=\{w_1^+, w_2^+, ..., w_n^+\},\ w_j^+=\max\{d(A_i)\times n_{ij}; \forall i \in \{1, 2, ..., n\}\}. \quad (16)$$

$$\text{NIS } A^-=\{w_1^-, w_2^-, ..., w_n^-\},\ w_j^-=\min\{d(A_i)\times n_{ij}; \forall i \in \{1, 2, ..., n\}\}. \quad (17)$$

(4) Calculate the distances: The distances from each alternative to *PIS* can be measured as follows:

$$d_i^+=\sqrt{\sum_{j=1}^{m}(w_{ij}^*-w_j^+)^2}, \quad i=1, 2, ..., n. \quad (18)$$

The distances from each alternative to *NIS* can be measured as follows:

$$d_i^-=\sqrt{\sum_{j=1}^{m}(w_{ij}^*-w_j^-)^2}, \qquad i=1, 2, ..., n \quad (19)$$

(5) Obtain the Ranks: The closeness coefficient given by,

$$CC_i=\frac{d_i^-}{d_i^+ + d_i^-}. \quad (20)$$

can be used to rank the countries.

III. METHODOLOGY

The selected factors for this study are population density (PD), age 65 or older (AGE), gross domestic product per capita (GDP), cardiovascular death rate (CDR), diabetes mellitus prevalence (DIA), hand washing facilities (HWF), hospital beds availability (HB) and life expectancy (LE). An internet-based survey is carried out up to date 23rd of September 2020. Through these e-sources the very high, high, moderate, low, very low risky COVID-19 spreading countries are identified [25]. Fifteen different countries are selected to compare the transmission of COVID-19. These countries are USA, Afghanistan, Colombia, Pakistan, Philippines, South Africa, India, Sri Lanka, China, Laos, Thailand, Timor, Mongolia, Vietnam and Yemen.

The pair-wise comparison matrix of risk factors is determined after reviewing the available literature [[6]- [11]]. To compare the factors, linguistic scales are used. The selected linguistic scales and their triangular fuzzy representation are in Table 1. Using this matrix, weights of

the factors are estimated using fuzzy extent analysis method [24]. Then obtained weights are compared, and the degrees of possibilities are calculated. Using these degrees of possibilities weight vector of the risk factors is derived. Finally, the ranks of the countries are obtained using TOPSIS method.

Table 1 Selected Linguistic scales to construct the pair-wise comparison matrix.

Linguistic scale	Triangular fuzzy numbers
Absolutely more important	(5/2, 3, 7/2)
Very strongly more important	(2, 5/2, 3)
Strongly more important	(3/2, 2, 5/2)
Weakly more important	(1, 3/2, 2)
Equally important Just equal	(1/2, 1, 3/2) (1, 1, 1)

IV. Results and Discussion

Step 1: Compare the factors and construct pair-wise comparison matrix

After reviewing the available facts [[6] – [11]] pair-wise comparison matrix which is shown in Table A1 (Refer Appendix) is constructed.

Step 2: Calculate the synthetic extent values

Deploying extent analysis method in (9), the calculated fuzzy synthetic extent values are given below.

=--=-

After comparing above values using (10) the following degrees of possibilities are found.

$$D(S_{PD} \geq S_{AGE}) = 0.93$$
$$D(S_{PD} \geq S_{GDP}) = 1$$
$$D(S_{PD} \geq S_{CDR}) = 1$$
$$D(S_{PD} \geq S_{DIA}) = 1$$
$$D(S_{PD} \geq S_{HWF}) = 1$$
$$D(S_{PD} \geq S_{HB}) = 1$$
$$D(S_{PD} \geq S_{LE}) = 1$$

$$D(S_{GDP} \geq S_{PD}) = 0.77$$
$$D(S_{GDP} \geq S_{AGE}) = 0.69$$
$$D(S_{GDP} \geq S_{CDR}) = 1$$
$$D(S_{GDP} \geq S_{DIA}) = 1$$
$$D(S_{GDP} \geq S_{HWF}) = 1$$
$$D(S_{GDP} \geq S_{HB}) = 1$$
$$D(S_{GDP} \geq S_{LE}) = 1$$

$$D(S_{DIA} \geq S_{PD}) = 0.3$$
$$D(S_{DIA} \geq S_{AGE}) = 0.2$$
$$D(S_{DIA} \geq S_{GDP}) = 0.5$$
$$D(S_{DIA} \geq S_{CDR}) = 0.63$$
$$D(S_{DIA} \geq S_{HWF}) = 1$$
$$D(S_{DIA} \geq S_{HB}) = 1$$
$$D(S_{DIA} \geq S_{LE}) = 1$$

$$D(S_{AGE} \geq S_{PD}) = 1$$
$$D(S_{AGE} \geq S_{GDP}) = 1$$
$$D(S_{AGE} \geq S_{CDR}) = 1$$
$$D(S_{AGE} \geq S_{DIA}) = 1$$
$$D(S_{AGE} \geq S_{HWF}) = 1$$
$$D(S_{AGE} \geq S_{HB}) = 1$$
$$D(S_{AGE} \geq S_{LE}) = 1$$

$$D(S_{CDR} \geq S_{PD}) = 0.64$$
$$D(S_{CDR} \geq S_{AGE}) = 0.55$$
$$D(S_{CDR} \geq S_{GDP}) = 0.89$$
$$D(S_{CDR} \geq S_{DIA}) = 1$$
$$D(S_{CDR} \geq S_{HWF}) = 1$$
$$D(S_{CDR} \geq S_{HB}) = 1$$
$$D(S_{CDR} \geq S_{LE}) = 1$$

$$D(S_{HWF} \geq S_{PD}) = 0$$
$$D(S_{HWF} \geq S_{AGE}) = 0$$
$$D(S_{HWF} \geq S_{GDP}) = 0$$
$$D(S_{HWF} \geq S_{CDR}) = 0.125$$
$$D(S_{HWF} \geq S_{DIA}) = 0.43$$
$$D(S_{HWF} \geq S_{HB}) = 1$$
$$D(S_{HWF} \geq S_{LE}) = 1$$

$$D(S_{HB} \geq S_{PD}) = 0$$
$$D(S_{HB} \geq S_{AGE}) = 0$$
$$D(S_{HB} \geq S_{GDP}) = 0$$
$$D(S_{HB} \geq S_{CDR}) = 0$$
$$D(S_{HB} \geq S_{DIA}) = 0$$
$$D(S_{HB} \geq S_{HWF}) = 0.6$$
$$D(S_{HB} \geq S_{LE}) = 1$$

$$D(S_{LE} \geq S_{PD}) = 0$$
$$D(S_{LE} \geq S_{AGE}) = 0$$
$$D(S_{LE} \geq S_{GDP}) = 0$$
$$D(S_{LE} \geq S_{CDR}) = 0$$
$$D(S_{LE} \geq S_{DIA}) = 0$$
$$D(S_{LE} \geq S_{HWF}) = 0$$
$$D(S_{LE} \geq S_{HB}) = 0$$

By using (11), following values are derived.

$$D'(A_{PD}) = D(S_{PD} \geq S_{AGE}, S_{GDP}, S_{CDR}, S_{DIA}, S_{HWF}, S_{HB}, S_{LE}) = 0.93$$
$$D'(A_{AGE}) = D(S_{AGE} \geq S_{PD}, S_{GDP}, S_{CDR}, S_{DIA}, S_{HWF}, S_{HB}, S_{LE}) = 1$$
$$D'(A_{GDP}) = D(S_{GDP} \geq S_{PD}, S_{AGE}, S_{CDR}, S_{DIA}, S_{HWF}, S_{HB}, S_{LE}) = 0.69$$
$$D'(A_{CDR}) = D(S_{CDR} \geq S_{PD}, S_{AGE}, S_{GDP}, S_{DIA}, S_{HWF}, S_{HB}, S_{LE}) = 0.55$$
$$D'(A_{DIA}) = D(S_{DIA} \geq S_{PD}, S_{AGE}, S_{GDP}, S_{CDR}, S_{HWF}, S_{HB}, S_{LE}) = 0.2$$
$$D'(A_{HWF}) = D(S_{HWF} \geq S_{PD}, S_{AGE}, S_{GDP}, S_{CDR}, S_{DIA}, S_{HB}, S_{LE}) = 0.125$$
$$D'(A_{HB}) = D(S_{HB} \geq S_{PD}, S_{AGE}, S_{GDP}, S_{CDR}, S_{DIA}, S_{HWF}, S_{LE}) = 0$$
$$D'(A_{LE}) = D(S_{LE} \geq S_{PD}, S_{AGE}, S_{GDP}, S_{CDR}, S_{DIA}, S_{HWF}, S_{HB}) = 0$$

Then the derived weight vector for the selected factors is given by,

$$W' = (0.93, 1, 0.69, 0.55, 0.2, 0.125, 0, 0)^T$$

Using (13) following normalized weight vector is derived,

$$W = (0.34, 0.37, 0.26, 0.204, 0.074, 0.046, 0, 0)^T$$

Factor wise distribution of these weights are graphically shown in Figure 1.

According to Figure 1, the risk factor "Age" takes the highest weight and the second, third, fourth, fifth and sixth places are taken by the factors "population density", "GDP", "cardiovascular death rate", "diabetes prevalence" and handwashing facilities respectively. The factors "hospital beds" and "life expectancy" takes zero weights. Therefore, from now onwards they have not considered as the significant risk factors in the study.

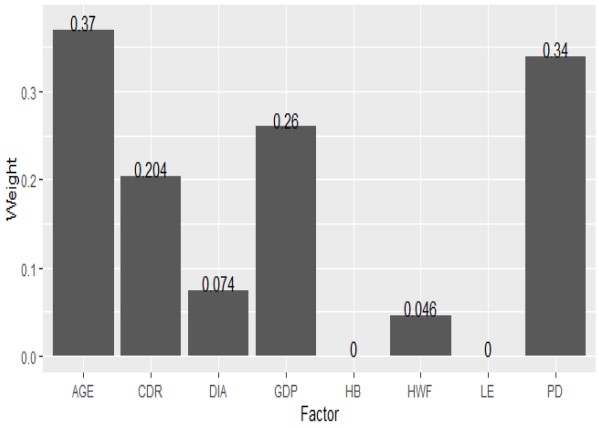


Fig 15 Plot of weights of selected factors

Table 2 . Normalized weighted decision matrix

Country	PD	Age 65 or Older	GDP	CDR	DIA	HWF
United States	0.0149	0.1946	0.2151	0.0235	0.0262	0.0070
Afghanistan	0.0227	0.0326	0.0072	0.0930	0.0233	0.0167
Colombia	0.0185	0.0965	0.0526	0.0194	0.0180	0.0097
Pakistan	0.1067	0.0567	0.0200	0.0659	0.0202	0.0106
Philippines	0.1469	0.0606	0.0301	0.0577	0.0171	0.0080
South Africa	0.0195	0.0675	0.0488	0.0312	0.0134	0.0144
India	0.1880	0.0756	0.0255	0.0440	0.0252	0.0106
Sri Lanka	0.1427	0.1271	0.0463	0.0307	0.0259	0.0069
China	0.0616	0.1343	0.0607	0.0408	0.0236	0.0120
Laos	0.0124	0.0509	0.0254	0.0573	0.0097	0.0127
Thailand	0.0564	0.1436	0.0646	0.0171	0.0171	0.0070
Timor	0.0364	0.0449	0.0261	0.0522	0.0166	0.0224
Mongolia	0.0008	0.0509	0.0470	0.0717	0.0117	0.0089
Vietnam	0.1286	0.0903	0.0245	0.0382	0.0145	0.0074
Yemen	0.0223	0.0369	0.0059	0.0771	0.0130	0.0127

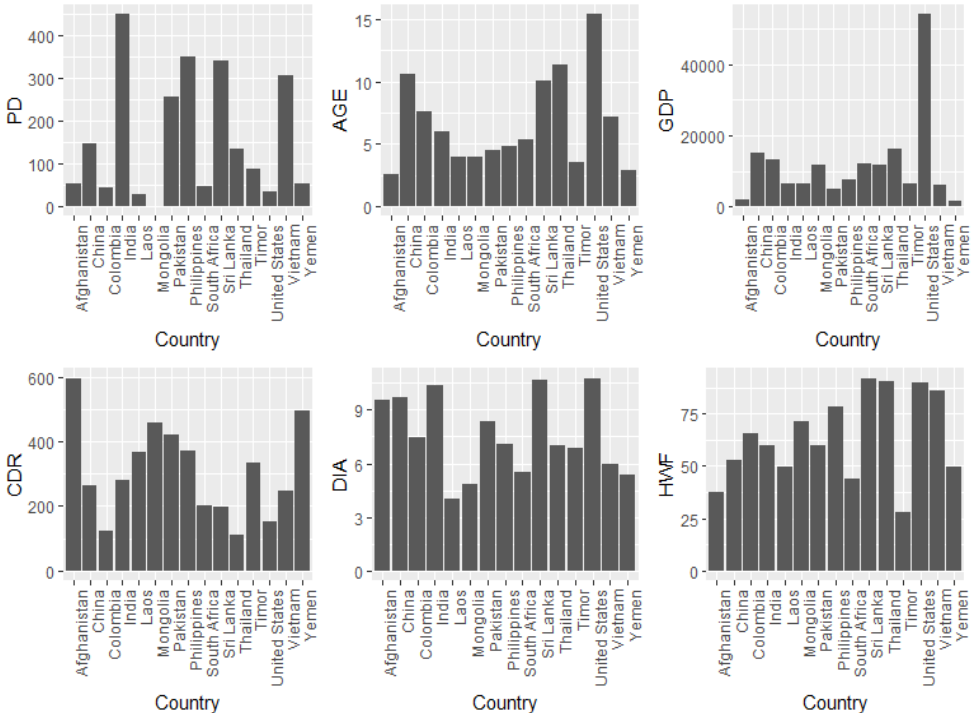


Figure 16 Factor distribution of selected countries

According to the Figure 2, it depicts how the factors distribute in each selected country. Among the selected countries India shows the highest population density and United States has the highest elder population as well as the highest GDP. Afghanistan has the highest cardiovascular deaths rate and United states has the highest diabetes prevalence techniques. Sri Lanka also shows a tie with United States in preventing diabetes. Finally, moving to the handwashing facilities, Sri Lanka, United States as well as Thailand and Yemen are notable.

Step 3: Find the ranks of the countries

Factor distributions of the countries are shown graphically in Figure 2. After multiplying each normalized value of factors with their weights, Table 2 is constructed.

Using (16) derived PIS A^+ vector is

[0.1880, 0.1946, 0.2151, 0.0930, 0.0262, 0.0224]. Using (17) derived NIS A^- vector is [0.0008, 0.0326, 0.0059, 0.0171, 0.0097, 0.0069]. The solution distance of each country from PIS A^+ and NIS A^- are measured with the aid of (18) and (19). Finally, the ranks are obtained using (20). Table 3 shows the selected countries with their corresponding ranks.

The Centers of Disease Control and Prevention (CDC) [25] prohibited some countries to travel according to the spread of COVID-19 categorized the risk as very high, high, moderate, low and very low. They have noticed that United States, Afghanistan, Colombia, Pakistan, Philippines, South Africa, India, China, Sri Lanka, Vietnam and Yemen as high risky COVID-19 spreading countries. Also, Mongolia as a moderately COVID-19 spreading country. Thailand as a low spreading country and Laos and Timor as very low COVID-19 spreading countries.

According to the final ranking the countries Timor and Laos ranked as 14 and 15 and it tallies with the CDC given information. Even though, Thailand was under low COVID-19 spreading category, it has taken the 5th place. Considering the country details, it can be decided that their adult population (age above 65+) are similar to the adult population in China and Sri Lanka. Therefore, it may cause to rank it in the 5th place. Mongolia is ranked in the 11th place and CDC also categorized it as a moderately COVID-19 spreading country. So, it matches with the CDC specification. United States, Afghanistan, Colombia, Pakistan, Philippines, South Africa, India, China, Sri Lanka, Vietnam and Yemen are ranked in the places of 1st, 10th, 9th, 8th, 4th, 12th, 3rd, 6th, 2nd, 7th and 13th. According to the current world situation United States has reported the highest number of coronavirus cases. Considering the data set it can be observed that the USA has the highest adult population among the selected countries. Also, it has the highest GDP and its CDR is also highly noticeable. So, concerning these facts we can say it is at the top. Even though Sri Lanka got the 2nd place, as a highly COVID-19 spreading country we are able to control this disease as the government took the relevant actions by making the public aware about the horrendousness of this pandemic. According to the world's statistics [26] it is confirmed that India is the country which has second most reported COVID-19 cases. Considering the dataset, it can be concluded that India has low handwashing facilities compared to US and Sri Lanka. So, it could be a reason to have many more COVID-19 cases inside the country.

Considering about Yemen and Afghanistan we can notice that their adult population is very low comparing to the other countries. It may have caused them to have 13th and 10th places even they are categorized under the high risk COVID-19 countries. Philippines and Pakistan have a higher population density and also currently it is identifying as a fast COVID-19 spreading country. Sometimes the uncontrollable factor population density may cause to the fast spread. South Africa also became noticeable since the reported COVID-19 cases are higher than the other countries. Analyzing the data set it can be concerned that in South

Africa all the factors we have considered takes a moderate value compared to the other countries.

Table 3 Ranks of the countries

Country	No of Cases	CC_i Value	Rank
United States	6896274	0.5866	1
Afghanistan	39145	0.2060	10
Colombia	777537	0.2352	9
Pakistan	307418	0.3214	8
Philippines	291784	0.3990	4
South Africa	663282	0.1800	12
India	5646010	0.4601	3
Sri Lanka	3313	0.4712	2
China	90399	0.3831	6
Laos	23	0.1435	15
Thailand	3514	0.3840	5
Timor	27	0.1673	14
Mongolia	313	0.1955	11
Vietnam	1068	0.3808	7
Yemen	2032	0.1713	13

China is in a prominent place when taking about COVID-19. Their adult population is similar to Sri Lanka and their hand washing facilities are not at the level of Sri Lanka. But they are succeeded in controlling the virus due to their awareness about the hazardous of this epidemic and in this study the place they took tallies with the data set. Considering Vietnam and Colombia, Vietnam has a high population density compared to Colombia, but their handwashing facilities are rather impressive compared to Colombia.

All in all, the ranking of this study can be concluded that these results are agreed with the prior knowledge.

V. Conclusion

In this study a fuzzy mathematical approach is used to rank the selected countries according to the risk of the spread of COVID-19. Since COVID-19 is a novel virus, the researchers are investigating for factors which cause to the outbreak of this disease. Some countries are selected considering the spread of COVID-19 as vey high, high, moderate, low and very low and for each country the data are obtained for each selected factor. The selected factors are population density, age 65 or older, GDP, cardiovascular death rate, diabetes mellitus prevalence, hand washing facilities, hospital beds availability and life expectancy and the selected countries are United States, Afghanistan, Colombia, Pakistan, Philippines, South Africa, India, Sri Lanka, China, Laos, Thailand, Timor, Mongolia, Vietnam and Yemen.

To obtaine the results a fuzzy extent approach and TOPSIS are together used concerning the uncertainty and the impreciseness of the factors. Some data such as handwashing facilities are not available for most of the countries. So, it is a limitation of this study. Here we have taken only a limited number of countries. It is important to modify this analysis by adding more risk factors such as male/female smoking, poverty, human development index etc for a more accurate ranking.

References

[1] "Sri Lanka | COVID-19| coronavirus| WHO." https://www.who.int/srilanka/covid-19

[2] "Countries where Coronavirus has spread - Worldometer." https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/

[3] "Q&A on coronaviruses (COVID-19)." https://www.who.int/news-room/q-a-detail/q-a-coronaviruses

[4] "Coronavirus disease (COVID-19) – World Health Organization." https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[5] CDC, "Coronavirus Disease 2019 (COVID-19)," Centers for Disease Control and Prevention, Feb. 11, 2020. https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html

[6] "owid/covid-19-data," GitHub. https://github.com/owid/covid-19-data

[7] R. Singh and S. Avikal, "COVID-19: A decision-making approach for prioritization of preventive activities," International Journal of Healthcare Management, 13, 257–262 (2020)

[8] M. Arsalan, O. Mubin, F. Alnajjar, and B. Alsinglawi, "COVID-19 Global Risk: Expectation vs. Reality," Int. J. Environ. Res. Public Health, 10 (2020)

[9] "COVID-19 in Asia_ A Country-By-Country Guide – The Diplomat.pdf."

[10] C. Iwendi et al., "COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm," Front. Public Health, 8, 357 (2020)

[11] "Environmental and social analysis as risk factors for the spread of the novel coronavirus (SARS-CoV-2) using remote sensing, GIS and analytical hierarchy process (AHP) Case of Peru.pdf."

[12] E. M. L. Aquino et al., "Social distancing measures to control the COVID-19 pandemic: potential impacts and challenges in Brazil," Ciência & Saúde Coletiva, 25, 2423–2446 (2020)

[13] "Masks and social distancing help control Coronavirus, but hand washing and other measures still need to be followed - Health News, Firstpost."

https://www.firstpost.com/health/masks-and-social-distancing-help-control-coronavirus-but-hand-washing-and-other-measures-still-need-to-be-followed-8439261.html

[14] "New Report Ranks Countries by COVID-19 Safety," p. 3.

[15] "COVID-19 High risk groups." https://www.who.int/westernpacific/emergencies/covid-19/information/high-risk-groups

[16] J. Glanz et al., "Coronavirus Could Overwhelm U.S. Without Urgent Action, Estimates Say," The New York Times (2020)"

[17] Sri Lanka ranked 16th among high risk countries for COVID - 19.pdf."

[18] T. S. Piyatilake, S. S. N. Perera, and S. K. Boralugoda, "Developing a decision support index to quantify air quality: a mathematical modeling approach," 17 (2016)

[19] K. Kumar and S. Kumanan, "Decision Making in Location Selection: An Integrated Approach with Clustering and TOPSIS," 16 (2012)

[20] S. Murugaanandam and V. Ganapathy, "Reliability-Based Cluster Head Selection Methodology Using Fuzzy Logic for Performance Improvement in WSNs," IEEE Access, vol. 7, 87357–87368 (2019)

[21] W. Wu, Z. Xu, G. Kou, and Y. Shi, "Decision-Making Support for the Evaluation of Clustering Algorithms Based on MCDM," Complexity, 1–17 (2020)

[22] T. P. Swindiarto, R. Sarno, and D. C. R. Novitasari, "Integration of Fuzzy C-Means Clustering and TOPSIS (FCM-TOPSIS) with Silhouette Analysis for Multi Criteria Parameter Data," in 2018 International Seminar on Application for Technology of Information and Communication, Semarang, 463–468 (2018)

[23] Y. Peng, Y. Zhang, G. Kou, and Y. Shi, "A Multicriteria Decision Making Approach for Estimating the Number of Clusters in a Data Set," PLoS ONE, vol. 7, no. 7, p. e41713, (2012)

[24] D.-Y. Chang, "Applications of the extent analysis method on fuzzy AHP," European Journal of Operational Research, 95 649–655 (1996)

[25] CDC, "Coronavirus Disease 2019 (COVID-19)," Centers for Disease Control and Prevention, Feb. 11, 2020. https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-with-medical-conditions.html

[26] "Countries where Coronavirus has spread - Worldometer." https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/

[27] D. S. Ediriweera, N. R. de Silva, N. G. Malavige, and H. J. de Silva, "AN epidemiological model to aid decision-making for covid-19 control in sri lanka," medrxiv, p. 2020.04.11.20061481

APPENDIX

Table A1. Pairwise comparison matrix

	PD	AGE	GDP	CDR	DIA	HWF	HB	LE
PD	(1,1,1)	(1.75,2.25,2.75)	(1.75,2.25,2.75)	(2,2.5,3)	(2,2.5,3)	(2.25,2.75,3.25)	(2,2.5,3)	(2,2.5,3)
AGE	(0.39,1,0.7)	(1,1,1)	(2.25,2.75,3.25)	(2.5,3,3.5)	(2.5,3,3.5)	(2.5,3,3.5)	(2.5,3,3.5)	(2.5,3,3.5)
GDP	(0.37,0.45,0.58)	(0.31,0.37,0.45)	(1,1,1)	(2.25,2.75,3.25)	(2.25,2.75,3.25)	(2.25,2.75,3.25)	(2.5,3,3.5)	(2.25,2.75,3.25)
CDR	(0.34,0.42,0.53)	(0.29,0.33,0.4)	(0.31,0.37,0.45)	(1,1,1)	(2.5,3,3.5)	(2.5,3,3.5)	(2.5,3,3.5)	(2.5,3,3.5)
DIA	(0.34,0.42,0.53)	(0.29,0.33,0.4)	(0.31,0.37,0.45)	(0.29,0.33,0.4)	(1,1,1)	(2.5,3,3.5)	(2.5,3,3.5)	(2.5,3,3.5)
HWF	(0.31,0.37,0.45)	(0.29,0.33,0.4)	(0.31,0.37,0.45)	(0.29,0.33,0.4)	(0.29,0.33,0.4)	(1,1,1)	(2.25,2.75,3.25)	(2.25,2.75,3.25)
HB	(0.34,0.42,0.53)	(0.29,0.33,0.4)	(0.29,0.33,0.4)	(0.29,0.33,0.4)	(0.29,0.33,0.4)	(0.31,0.37,0.45)	(1,1,1)	(2.5,3,3.5)
LE	(0.34,0.42,0.53)	(0.29,0.33,0.4)	(0.31,0.37,0.45)	(0.29,0.33,0.4)	(0.29,0.33,0.4)	(0.31,0.37,0.45)	(0.29,0.33,0.4)	(1,1,1)

Programme Committee

Conference Chair

- Prof. Asoka S Karunananda, University of Moratuwa, Sri Lanka.

Technical Programme Chair

- Dr. (Mrs.) D.D.M. Ranasinghe, The Open University of Sri Lanka, Sri Lanka.

Conference Secretary

- Dr. (Mrs.) N. M. Wagarachchi, University of Ruhuna, Sri Lanka.

.

Conference Treasurer

- Dr. (Mrs.) W.A.C. Weerakoon, University of Kelaniya, Sri Lanka.

Organizing Committee

- Dr. R. M. Kapila Tharanga Rathnayaka, Sabaragamuwa University of Sri Lanka.
- Dr. S Vasanthapriyan, Sabaragamuwa University of Sri Lanka.
- Dr. Sagara Sumathipala, University of Moratuwa, Sri Lanka
- Dr. Budditha Hettige, General Sir John Kotelawala Defence University, Sri Lanka.
- Mrs. DU Vidanagama, General Sir John Kotelawala Defence University, Sri Lanka
- Mr. HWH Premachandra, Wayamba University of Sri Lanka, Sri Lanka

Panel of Reviewers

- Prof. G Rzevski, The Open University, United Kingdom
- Prof. AS Karunananda, University of Moratuwa, Sri Lanka
- Prof. ND Kodikara, University of Colombo School of Computing, Sri Lanka
- Prof. NGJ Dias, University of Kelaniya, Sri Lanka
- Prof. S Vasanthapriyan, University of Sabaragamuwa, Sri Lanka
- Dr. S Jayalal, University of Kelaniya, Sri Lanka
- Dr. (Mrs.) D.D.M. Ranasinghe, The Open University, Sri Lanka
- Dr. (Mrs.) Hemali Ratnayake, The Open University, Sri Lanka
- Dr. L Ranathunga, University of Moratuwa, Sri Lanka
- Dr. DD Karunarathna, University of Colombo School of Computing, Sri Lanka
- Dr. A Atukorale, University of Colombo School of Computing, Sri Lanka
- Dr. C Keppitiyagama, University of Colombo School of Computing, Sri Lanka
- Dr. TMKK Jinasena, University of Sri Jayawardenapura, Sri Lanka
- Dr. (Mrs.) A. Ariyaratne, University of Sri Jayawardenapura, Sri Lanka
- Dr. RD Nawarathna, University of Peradeniya, Sri Lanka
- Dr. TGI Fernando, University of Sri Jayawardenapura, Sri Lanka
- Dr. (Mrs.) KSD Fernando, University of Moratuwa, Sri Lanka
- Dr. (Ms.) GU Ganegoda, University of Moratuwa, Sri Lanka
- Dr. S Liyanage, University of Kelaniya, Sri Lanka
- Dr. (Mrs.) ATP Silva, University of Moratuwa, Sri Lanka
- Dr. C Rajapakse, University of Kelaniya, Sri Lanka
- Dr. (Mrs.) T Sandanayake, University of Moratuwa, Sri Lanka
- Dr. S Sumathipala, University of Moratuwa, Sri Lanka
- Dr. PKDDP Pitigala, University of Sri Jayawardnapura, Sri Lanka

- Dr. SP Pitigala, University of Kelaniya, Sri Lanka
- Dr. KJC Kumara, University of Ruhuna, Sri Lanka
- Dr. Lalith Premaratne, General Sir John Kotelawala Defence University, Sri Lanka
- Dr. RMKT Rathnayaka, University of Sabaragamuwa, Sri Lanka
- Dr. LP Kalansooriya, General Sir John Kotelawala Defence University, Sri Lanka
- Dr. (Mrs.) NV Chandrasekara, University of Kelaniya, Sri Lanka
- Dr. (Mrs.) NM Wagarachchi, University of Ruhuna, Sri Lanka
- Dr. (Mrs.) WAC Weerakoon, University of Kelaniya, Sri Lanka
- Dr. B Hettige, General Sir John Kotelawala Defence University, Sri Lanka
- Mrs. DU Vidanagama, General Sir John Kotelawala Defence University, Sri Lanka
- Mr. R Bandara, University of Sri Jayawardenapura, Sri Lanka
- Mrs. WPJ Pemarathne, General Sir John Kotelawala Defence University, Sri Lanka
- Mr. DS Deegalla, University of Peradeniya, Sri Lanka
- Mr. D Perera, University of Sri Jayawardenapura, Sri Lanka

Compering Team

- Dr. (Mrs.) WAC Weerakoon, University of Kelaniya, Sri Lanka
- Mrs. DU Vidanagama, General Sir John Kotelawala Defence University, Sri Lanka
- Mrs. ADD De Silva, University of Kelaniya, Sri Lanka