



University of Computer Studies (Mandalay)

Department of Advanced Science and Technology

Ministry of Science and Technology

The Republic of the Union of Myanmar

JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

June, 2023

JiTRi 2023

JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

June, 2023

Organized by

University of Computer Studies

(Mandalay)

Volume-3, Issue-1

JOURNAL OF INFORMATION TECHNOLOGY, RESEARCH AND INNOVATION

JITRI, 2023

Vol-3, Issue-1

UNIVERSITY OF COMPUTER STUDIES (MANDALAY)

Journal of Information Technology, Research and Innovation (JITRI), 2023 Committee

University of Computer Studies (Mandalay)

Editor in Chief *Prof. Dr. San San Tint, Pro Rector*

Technical Program Committee & Editorial Board

- Prof. Dr. Thin Thin Naing
- Prof. Dr. Aye Aye Chaw
- Prof. Dr. Thandar Aung
- Prof. Dr. Mya Thida Kyaw
- Prof. Dr. Yu Ya Win
- Prof. Dr. Saw Thandar Myint
- Prof. Dr. May Mya Aung
- Assoc. Prof. Dr. Thein Htay Zaw
- Assoc. Prof. Daw Yin Min Tun
- Assoc. Prof. Daw San San Lwin
- Dr. May Phyo Thu

Acknowledgements for Reviewers and Organizing members of JITRI

The Journal of Information Technology, Research and Innovation (JITRI) 2023 technical program committee would like to express the deepest appreciation to the external reviewers and organizing members for collaborating to publish this journal successfully. JITRI (2023) is published as the third time in order to undertake research in their respective fields of studies. Finally, we would like to thank all the authors who had submitted their research papers by making their great efforts in the Journal of Information Technology, Research and Innovation (JITRI), 2023, Vol- 3, Issue-1.

Table of Contents

FAILURE LOG ANALYSIS USING MACHINE LEARNING TECHNIQUES

Su Lae Lae Hnin[1] and Zin Mar Myo[2]

University of Computer Studies (Magway)
sulae.ucsmgy@gmail.com, zinmarmyo@ucsmgy.edu.mm

ABSTRACT

The log data that is produced automatically by the system stores the information about the events that have taken place in the system. Log files thus contain highly valuable information about the larger computer system. One of the challenges of designing and deploying a large system such as IBM's BlueGene/L which has 128K processors, is the need to easily know which events are failure and to trace back which component the failure log events happened on. By using the log files, failure logs can be characterized by analyzing relevant logs. However, due to the very high number of log entries of some systems, it is difficult to detect failure logs. Therefore, machine learning and data mining techniques are widely used for analyzing logs. The aim of this paper is to predict failure log events by using k-means clustering, and the places where these failure events occur are analyzed. For evaluating the result of this methodology, this paper applies rule-based classification. The experiments are taken on the public dataset from Blue Gene/L (BGL) supercomputer.

KEYWORDS

log files, machine learning, clustering, k-means, failure log events, Blue Gene/L

1. INTRODUCTION

The usage events of a system in the form of log files can be used for troubleshooting. Log files are a historical record of everything and anything that happens within a system, including events such as transactions, errors and intrusions [1]. The system logs record all the errors, warnings, notifications and information that comes from the embedded software that runs on the products (or hardware). The log messages are usually semi-structured text strings. These can be used to record the events or state of interests. However, the log files can be different types and format. Therefore, it is not easy to understand whether the individual log event is failure or not. Initially, the developer or system administrator manually detect the log event as failure or exception by searching keywords or regular expression matching according to their relevant field knowledge. Actually, this manual detection can be time consuming and high error rate. Therefore, many machine learning techniques are applied in predicting the failure log events [2][3][4].

But an automated log analysis is still an ongoing challenge. Some reasons for this:

- the sheer rates at which log events are generated,
- log messages have different types and format, and
- log files are usually unlabeled data. Thus, unsupervised methods to detect anomalies in logs are the trend.

Thus, we analyze the log file for predicting the failure events. Our work is based on clustering technique which is unsupervised learning. The procedures of this work is as follows:

- Data preprocessing (removing repetition, and cleaning),
- Creating weight matrix by using TF-IDF (Term Frequency–Inverse Document Frequency),
- Clustering the log events by using k-means
- Identifying the log event that can be said as failure, and where those log events are common
- Giving the evaluation result with different k values for k-means.

The rest of this paper is organized as follows. The related work is presented in section 2. Section 3 describes the system methodology. In section 4, evaluation results are presented, and section 5 wraps the system.

2. RELATED WORKS

Not only statically analysis on the log files, but a variety of data mining and machine learning algorithms are also used to analyze the information in the log files. Log analysis has been carried out in a variety of ways including anomaly detection, failure diagnosis, and behavior of logs of user.

Ali Hussain SHAH et al. [2] proposed automated anomaly detection method that combines unsupervised and supervised machine learning and domain knowledge to predict accurately anomalous log events from BGL. Firstly, they made the removal of digits and punctuation, keeping only textual data, as the pre-processing of data. Then clustering was done using k-means. These clusters were analyzed to create labelled dataset, with the help of domain knowledge. On these labelled dataset, classification is made to predict the log events as anomaly or normal.

Jin Wang et al. [3] exploited the unsupervised clustering to detect the anomaly without log parsing. The word-sequence weight matrix is created using TF-IDF. The feature vector of the log sequence is obtained from word vector generated by Word2Vec and the word sequence weight matrix. In this paper, they used only description part of the log messages from the BGL dataset. They approved their approach can effectively carry out unsupervised anomaly detection.

Yinglung Liang et al. [5] analyzed logs from Blue Gene/L prototype at IBM Rochester using a temporal-spatial filtering algorithm. They predicted the failure events (memory failure, network failure, application I/O failure, and so on) based on the failure characteristics. The strong correlations between the occurrence of a failure and several factors, including the time stamp of other failures, the location of other failures, and even the occurrence of non-fatal events, were found.

Yinglung Liang et al. [6] extracted a set of features that can accurately capture the characteristics of failures. On these failure events, they exploited four classifiers including RIPPER (a rule-based classifier), Support Vector Machines (SVMs), a traditional Nearest Neighbor, and a customized Nearest Neighbor for predicting failure events. As the evaluation measurement, they used precision, recall, and F-measure for the four classifiers.

3. PROPOSED SYSTEM

In this section, we present the system flow of our approach, which predicts the failure log events. The flow chart of the system is as shown in Figure 1.

3.1. Data Collection

We have used public log dataset collected from Blue Gene/L (BGL) supercomputer, housed at IBM Rochester, which currently stands at number 8 in the Top 500 list of supercomputers [8]. It has 4096 compute chips, with each chip containing two processors. The number of logs in BGL are totally 4,747,963 during six months (from Jun 2005 to January 2006). We consider the log events collected during the 1st month.

There are 1,048,042 log events as shown in Table 1.

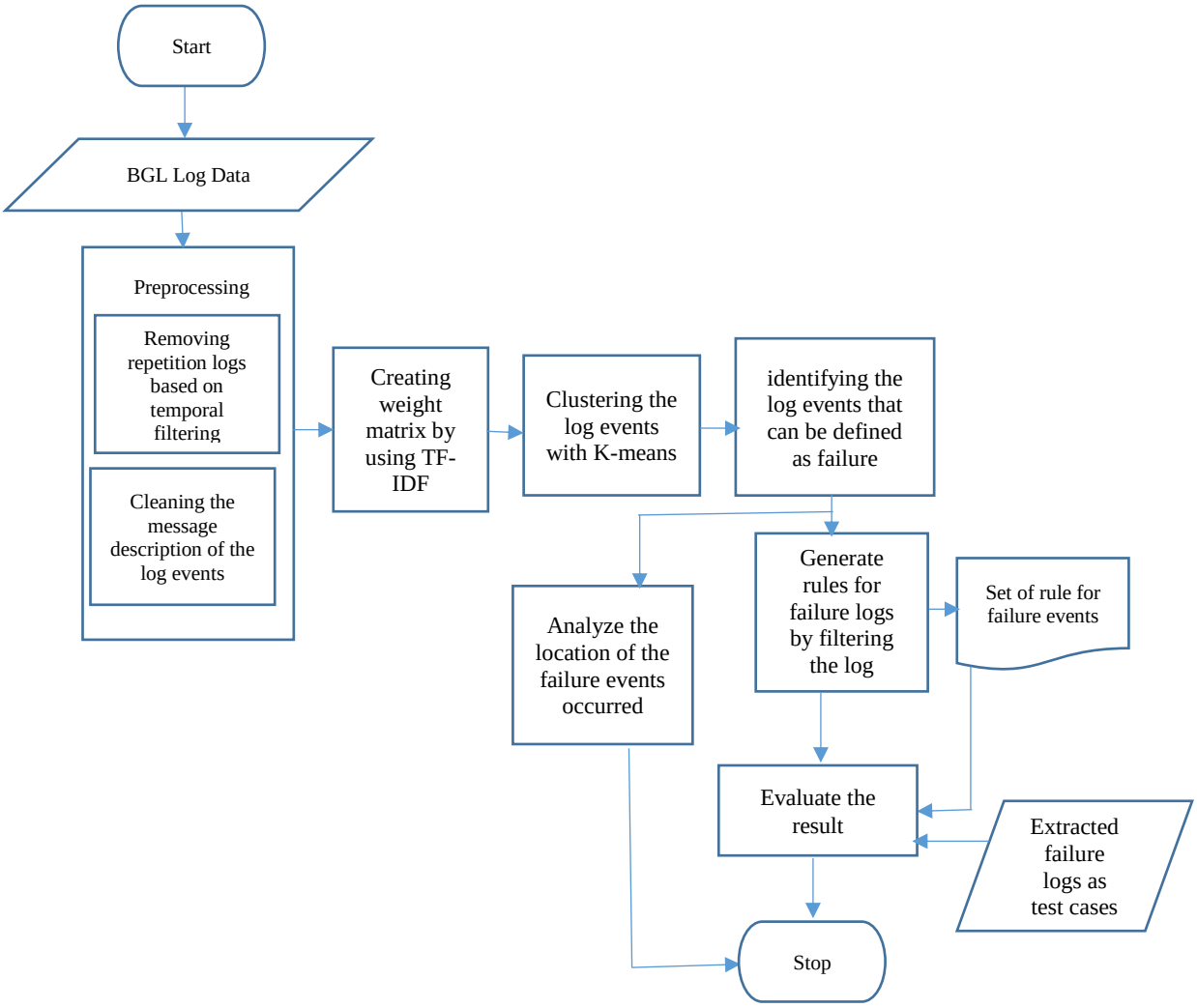


Figure 1. Flow of the Proposed System

Table 1. Description of Dataset

System	Start Date	End Date	Size	Number of log events
Blue Gene/L	2005-06-03	2005-07-03	145MB	1,048,042

Table 2. Event Characteristics of GBL Logs

Record ID	The sequence number for an event record in the log
Event time	The start time of event
Location	The location where the event occurs
Event type	The mechanism through which the event is recorded
Facility	The component where the event is flagged, which can be one of the following: LINKCARD, APP, KERNEL, DISCOVERY, MMCS, or MONITOR. LINKCARD
Severity	The level of the log messages performed an alert indicator, which can be one of the following: INFO, WARNING, SEVERE, ERROR, FATAL, or FAILURE.
Message part	A brief overview of the event condition.

Each record of the logs contains the attributes as shown in Table 2 [6]. There are seven attributes or characteristic in the log event.

Sample for BGL Log

1125667112 2005.09.02 R61-M1-NC-I:J18-U11 2005-09-02-06.18.32.983916 R61-M1-NC-I:J18-U11 RAS APP FATAL ciod: LOGIN chdir

In this log event,

Record ID – 1125667112
Event time - 2005-09-02-06.18.32.983916
Location - R61-M1-NC-I:J18-U11
Event type - RAS
Facility - APP
Severity - FATAL
Message part - ciod: LOGIN chdir (/bglscratch/bwallen/ SWL/ SYS-CALLS/ testcases/ kernel/ syscalls/link) failed: No such file or directory

3.2 Preprocessing

There are two steps for the preprocessing of the data: performing the temporal filtering to remove the duplicate reports from the same location, and cleansing the data (Tokenization and Stop words removal).

Before processing the log events, the repeated log events are removed by the following definition according to [8].

A log message M_2 is said to be a repetition of an original message M_1 if and only if

i. M_2 has the same message part as M_1
ii. Both M_2 and M_1 are generated from the same source (same location)
iii. The time within which both M_1 and M_2 are sent to the log is less than 1 second.

The explanation for removing the repeated log events are described as follows.

Raw Logs

2005-06-03-16.01.51. 141964 Compute Node INFO total of ddr error detected corrected
2005-06-03-16.03.36.365777 Compute Node INFO LEDRAM error detected corrected
2005-06-03-16.03.36.384059 Compute Node INFO LEDRAM error detected corrected
2005-06-03-16.04.02.488127 Compute Node INFO LEDRAM error detected corrected
2005-06-03-16.04.02.384059 Compute Node INFO LEDRAM error detected corrected

↓ After removing duplicate log events

2005-06-03-16.01.51. 141964 Compute Node INFO total of ddr error detected corrected
2005-06-03-16.03.36.365777 Compute Node INFO LEDRAM error detected corrected
2005-06-03-16.04.02.488127 Compute Node INFO LEDRAM error detected corrected

In the above example, there are five events in the raw log file. After removing duplicate log events according to the definition, three log events remain.
After removing the duplicate events in the logs file of our system, totally 1338 log events of 1,048,042 remain. Table 3 shows the number of log events for each severity level.

Table 3. Number of Log Message per Severity

Severity	Number of messages after removing repetition
Error	24
Failure	56
Fatal	732
Info	458
Severe	38
Warning	30

In this paper, we will identify which message parts mean the failure log events, and then analyze where the failures related to messages usually occur. The severity level of either FATAL or FAILURE is referred to failure [6]. After removing the repetition log events, we therefore extracted the characteristics "location", "severity", and "message part" from the log events. For the location characteristics, particular code which represent a particular hardware component are defined such as compute node (R02-M1-N0-**C**:J12-U11), I/O node (R22-M0-N0-**I**:J18-U01), Service card (R22-M0-S), link card (R15-M0-**L**0), node card (R00-M0-**N**0), a midplane in a rack (R76-**M**0)[8].

Then, number (e.g., IP, Port, and MAC addresses), punctuations, stop words (is, are, the, as, up, with, can, of, for, etc.) and non-alphanumeric characters (Example, **@, %, #**,) are removed from each log record. Finally, all of the text in the log files are transformed into lowercase letter to avoid case-sensitivity problems.

3.3 Creating Weight Matrix using TF-IDF

As the log events are textual representation, numerical representation must be transformed by creating weight matrix after preprocessing steps have been done. For creating weight matrix, TF-IDF (Term Frequency–Inverse Document Frequency) method is used.

The TF (Term Frequency) is the frequency of a word (i.e., number of times it appears) in a document [2]. TF can be calculated as,

$$TF = \frac{number\ of\ times\ term\ t\ appears\ in\ a\ document}{the\ number\ of\ terms\ in\ the\ document} \quad (3.1)$$

Inverse Document Frequency (IDF) is a measure to find how important a word is in the dataset [2], which is calculated as follows:

$$IDF = log \frac{number\ of\ documents}{number\ of\ documents\ containing\ term\ t} \quad (3.2)$$

TF-IDF score for each term in the log files is as:

$$TF\_IDF = TF * IDF \quad (3.3)$$

As an example, let's see the following sample log events, totally five log records.

1. compute info instruction cache parity correct
2. io fatal instruction cache parity correct
3. node card error idoproxydb hit assert expression source source line
4. node card warning ciod error
5. link severe not get assembly

The weight vale or numerical representation for the 3rd log event is

0.050 0.087 0.087 0.087 0.087 0.087 0.175 0.087

3.4 Clustering the Log Events with K-Means

As our log dataset which is used in the proposed system is unlabeled, k-means clustering, unsupervised learning is exploited to identify the failure log events. Clustering is used to identify the group of similar objects in the dataset with two or more variable quantities.

Steps of k-means algorithm [10]:

- Choose randomly k data objects from given dataset which works as centroid for k clusters initially.
- Compute distance of each data object from k centroids and then allocate each data object to the closest cluster with minimum centroid distance.
- Compute new centroid for each cluster by taking mean of the all data objects belonging to particular cluster. Calculate the total mean-square quantization error function. If error function reduces from previous one than these centroids will work as new centroids.

- Repeat step 2 and 3 until error function get constant.

3.5 Identifying the Failure Log Events and Generating Rules

After clustering the log events, the clusters that contains only the log events with severity level of failure and fatal are extracted, and these log events are defined as failure. In addition, the locations of these failure log events occur are also analyzed. Moreover, the rules are generated for evaluating the result. There are 290 failure log events from the k-means result with k=4. For these 290 failure log events, 16 types of rules can be generated. By using rule based classification, the predicted results are evaluated. Example of rule based classification is

If (location is compute_node and message part is ciod error load home bin invalid miss program), then log event is failure

4. EXPERIMENTAL RESULT

In this section, the experimental result is presented. Firstly, the rules that can be defined as failure logs are generated as the output of k-means with k=4.

Secondly, the analysis for the locations that these failure log messages usually occur is presented. Thirdly, the predicted results are evaluated.

4.1 Failure Log Events

As the result of k-means with k=4, there totally 290 log events that can be defined as failure. Then, by filtering the duplicate log events among 290 log events, we generate the rules for predicting the failure logs. There are totally 16 types of rule for failure log as shown in Table 4.

Table 4. Rules of Failure Logs for k=4

Location	Severity	message
compute_node	fatal	ciod error load home bin invalid miss program
io_node	fatal	ciod error load home bin invalid miss program
io_node	fatal	ciod error load stella raptor invalid miss program image file directory
location_null	failure	ciod exit abnormal dueto signal abort
io_node	fatal	ciod fail read message prefix control stream ciostream socket*
io_node	fatal	ciod loginchdir benchmark fail file directory
compute_node	fatal	ciod loginchdir benchmark fail file directory
compute_node	fatal	ciod loginchdir* fail file directory
compute_node	fatal	critical input interrupt enable
io_node	fatal	data tlb error interrupt
compute_node	fatal	data tlb error interrupt
compute_node	fatal	debug interrupt enable
compute_node	fatal	external input interrupt enable*
location_null	failure	idoproxy exit normal exit code*
location_null	failure	insert monitor info caught javalang point erexception
compute_node	fatal	machine check interrupt enable*

4.2 Analysis on the Source of Failure Log Events

For the 290 failure log records generated by the 4 clusters, the location that these failure log records occur is examined. We found that there are three types of location: compute node, I/O node, and location null. Location null means that there is no description in the location

characteristic of BGL log file. Without the log events that have NULL in location, there are 273 log records of 290. Of these 273 failure log events, 69% occurred at I/O node and 31% occurred at compute node. In the message parts of the predicted failure logs, which message parts are related to I/O node and which message parts are concerned to compute node are shown in Figure 2 and Figure 3 respectively.

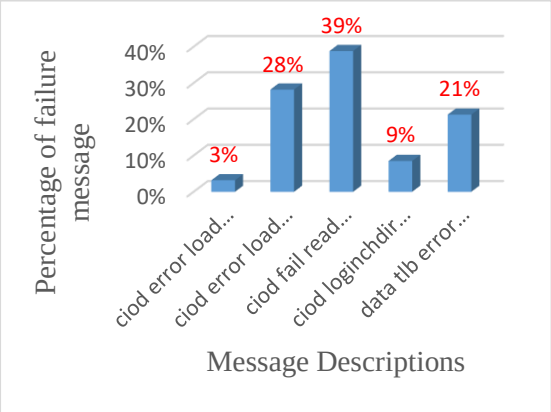


Figure 2. Failure Messages Occurrence at I/O Node

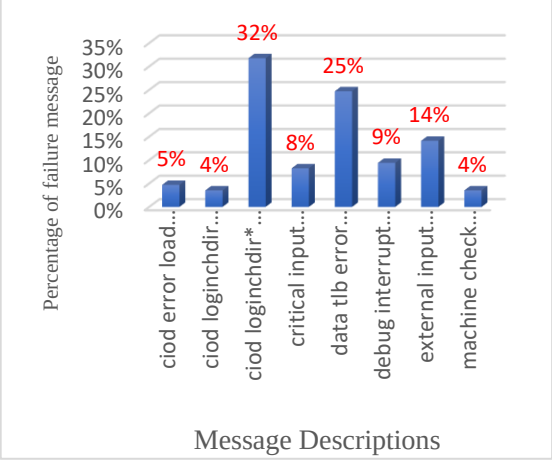


Figure 3. Failure Messages Occurrence at Compute Node

4.2 Result Evaluation

For evaluating the result, rule-based classification is applied. We use failure detection rate and outlier detection rate to measure the effectiveness of the system.

$$failure\ detection\ rate = \frac{number\ of\ records\ correctly\ classified\ by\ rule\ R}{total\ number\ of\ records} \quad (4.1)$$

$$outlier\ detection\ rate = \frac{number\ of\ records\ incorrectly\ classified\ by\ rule\ R}{total\ number\ of\ records} \quad (4.2)$$

From the first 100 log events out of 1338 log events, failure log events are extracted and used as test cases. There are 7 test cases as shown in Table 5. According to the results from k-means with k=2,3,4,5, and 6, the number of rules for predicting the failures logs are 8, 10,16, 22 and 23 respectively. The result evaluations are given on the different numbers of clusters k=2, 3, 4, 5, 6.

Table 5. Test Cases

location	Message	
io_node	ciod error load stella raptor invalid miss program image file directory	failure
io_node	ciod fail read message prefix control stream ciostream socket*	failure
io_node	ciod loginchdir benchmark fail file directory	failure
compute_node	data read plb error*	failure
io_node	data tlb error interrupt	failure
compute_node	data tlb error interrupt	failure
nod_card	temperature over limit link card	failure

Table 6. Result Evaluation

	K=2	K=3	K=4	K=5	K=6
Outlier detection rate	50%	50%	25%	25%	25%
Failure detection rate	50%	50%	75%	75%	75%

5. CONCLUSION

Log based analysis is important in assisting the system administrators for maintenance, diagnosis, and enhancing the overall system health. To address some challenges of the large system, such as BlueGene/L supercomputer, we propose the prediction system for the failure log events. The log data collected form this BlueGene/L system is used for our experiment. Firstly, the proposed system makes preprocessing and cleaning the data. After preprocessing step, the log data can be reduced by almost 99%. Then weight matrix of the data is created by using TF-IDF. Clustering approach, k-mean is then applied for predicting the log events that can be defined as failure. Next, the locations where these failure messages can occur are analyzed. These failure log messages usually occur at compute node, and I/O node. In result evaluation, the reason for detecting incorrectly the 25% of failure log messages is that the message parts in these logs can be not only fatal or failure severity, but also other severity types. The message parts in our prediction rules can be just fatal or failure severity. In addition, the optimum value of k can be concluded as 4 since the evaluation results have the same value starting from k=4 until k=6.

As the further extension, the domain area can be changed such as Thunderbird, Red Storm, Spirit and Liberty which are the world's most powerful supercomputers and this system can compare the results by using different system logs. This system can also be continued to implement with the supervise machine learning techniques.

REFERENCES

[1] Z. Zheng, L. Yu, W. Tang & Z. Lan (2011), "Co-analysis of RAS Log and Job Log on Blue Gene/P", *IEEE International Parallel & Distributed Processing Symposium*, pp840-851.

[2] A. H. Shah, D. Pasha, E. H. Zadeh & S. Konur (2022), "Automated Log Analysis and Anomaly Detection Using Machine Learning", *Fuzzy Systems and Data Mining* VIII, pp137-147.

[3] J. Wang, C. Zhao, S. He, Y. Gu, O. Alfarraj & A. Abugaba h (2022), "LogUAD: Log Unsupervised Anomaly Detection Based on Word2Vec", *Article-Computer Systems Science & Engineering*, Vol.41, No. 3.

[4] M. Siwach & S. Mann (2022), "Anomaly Detection for Web Log Data Analysis: A Review", *JOURNAL OF ALGEBRAIC STATISTICS*, Vol. 13, No.1, pp. 129-148.

[5] Y. Liang, Y. Zhang & M. Jette (2006), "BlueGene/L Failure Analysis and Prediction Models", *Proceedings of the 2006 International Conference on Dependable Systems and Networks.*

[6] Y. Liang, Y. Zhang & H. Xiong (2007), "Failure Prediction in IBM BlueGene/L Event Logs", *Seventh IEEE International Conference on Data Mining*.

[7] Blue Gene Team (2002), "An Overview of the BlueGene/L Supercomputer", *IEEE*.

[8] N. Taerat, N. Naksinehaboon, C. Chandler & J. Elliott (2009), "Blue Gene/L Log Analysis and Time to Interrupt Estimation", *2009 International Conference on Availability, Reliability and Security*, Fukuoka, Japan, pp173-180.

[9] S. He, J. Zhu, P. He & M. Lyu(2016), "Experience Report: System Log Analysis for Anomaly Detection*", IEEE 27th International Symposium on Software Reliability Engineering*, pp207-218.

[10] L. Singh & A. sharm a (2016), "Analysis and Classification of Internet activity Logs Based on Patterns of Traffic Rates", *INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY (IRJET)*, Vol.3, Issue.06.

Authors

Biography

Su Lae Lae Hnin is, a Junior Immigration Assistance (Dagon Myothit-Lawaka), and a candidate of M.C.Sc, University of Computer Studies (Magway). She received B.C.Sc. degree at University of Computer Studies (Magway), in 2018.

Zin Mar Myo, Associate Professor at Faculty of Computer Science, University of Computer Studies (Magway). She received M.C.Sc. degree at University of Computer Studies (Mandalay), in 2009, and Ph.D. degree at the University of Computer Studies (Yangon), in 2015. She has published 8 research papers.

SENTIMENT ANALYSIS OF COMMENTS IN MYANMAR LANGUAGE ON MOBILE DATA SERVICES

Nway Nway[1] and Zar Lwin Phyo[2]

[1,2]University of Computer Studies (Magway)
nwaynway13@ucsmgy.edu.mm, zarlwinphyo@ucsmgy.edu.mm

ABSTRACT

Nowadays, the rate of information technology is rising continuously and this is the reason why social media is well placed as a source of data to gather public opinions. Sentiment analysis is important to study people's emotions and opinions on given topics. Sentiment analysis in mobile data service comments analyzes the influence of data services of operators. Therefore, the proposed system performs the sentiment analysis to see the level of public satisfactions about data service of telecommunication operators in Myanmar. Using the Myanmar language comments about each telecommunication operator, this system explores public satisfactions about data services. Dataset is collected from comments of four operators (MPT, ATOM, Ooredoo, and Mytel) in Myanmar Language. Then the raw data comments are preprocessed. For sentiment analysis, this system uses the Multinomial Naive Bayes (MNB) Classifier with Chi-Square(X^2) goodness of fit test for feature selection method that reduces influential features. By using this system, operators of data service providers know the strengths and weaknesses about their services. This system suggests which operator has the best service among four operators in Myanmar. The results in this study obtained an average f1-score of 90%, precision 91%, recall 90% and 90% accuracy (Chi-Square significance level α 0.5).

KEYWORDS

Sentiment Analysis, Chi-square(X^2), MNB Classifier.

1. INTRODUCTION

With the explosive growth of social media (i.e., reviews, forums, blogs, and social networks) on the Internet, there is increased use of mass media in our daily lives by individuals or groups for their decision-making. People are using the Internet in recent years. In 2022, greater spending on social media (Facebook) increased by 37.5 percent of the population in Myanmar.

The proposed system only focuses on mining opinions that indicate positive, negative, or neutral sentiments. Businesses always want to seek public or consumer opinions about their services and products. Potential users also want to know the opinions of existing users before they use a service or purchase a product.

Every operator of mobile data service providers computes to provide the best Internet data services and to give advertisements or promotions to attract people. Actually, people do not want to choose wrong in using data service operators. Sentiment analysis of mobile data services reviews through social media proposes to see how much the level of community satisfaction in using data services operators (MPT, ATOM, Ooredoo, and Mytel) in Myanmar. Data is obtained from the official page of each operator on Facebook by using exportcomments.com. The results of sentiment analysis can be used by the operators of data service providers in improving their services and by the users in providing information to the community in order to choose a good mobile operator service.

For sentiment analysis, the proposed system uses a machine learning approach, Multinomial Naive Bayes (MNB) classifier, to determine the various opinions from given data services reviews on social media. This method supports the excellent performance for real-time data service comments and better results in terms of accuracy.

2. RELATED WORKS

Most of the research papers apply various methods for reviews or comments on social media in sentiment analysis or opinion mining. Most studies have been dedicated to English and other Latin languages. Nevertheless, there is no research on the Myanmar language for reviews or comments on social media.

Khin Thandar Nwet used machine learning algorithms such as Naive Bayes (NB), K-nearest neighbors (KNN), and Support vector machine (SVM) algorithms at making a comparative study between mentioned algorithms on a Myanmar data set [4]. Rimba Nuzulul Chory made a Support Vector Machine (SVM) algorithm to analyze the user satisfaction level of mobile data services [7]. H. H. Htay [3] proposed a 2-step longest matching approach. The first step, was syllable segmentation, in the second step left-to-right syllable longest matching forward segmentation was performed.

Dey et al [5] classified review datasets into positive and negative by using Naïve Bayes' and K-NN Classifier. Shahare et al [1] used Naïve Bayes and Levenshtein algorithm to find out emotion level the social media news data. This method gives better performance for real time news data on social media. Nurhayati et al [6] shows the effect of Chi-Square feature selection on the Naïve Bayes algorithm performance in analyzing sentiment document. Febriyani et al [2] used Naïve Bayes algorithm to classify sentiment on the level of customer satisfaction to Data Cellular Service.

Based on the results of research that has been done, this paper used Chi-Square feature selection and Naïve Bayes Classifier to classify comments in Myanmar language on mobile data service.

3. RESEARCH METHODOLOGY

The main goal of the proposed system is to analyze the comments of mobile data services on Facebook and to create a data service comment data set. In the proposed system, the first step is to collect the raw data from the official page of each operator on Facebook to develop a training corpus because of using the supervised learning approach. The next step is preprocessing after the data service reviews are collected. The steps in preprocessing include word segmentation, stemming, and stopwords removal. Moreover, to select feature words from the text of the opinions contained in the dataset, Chi-Square (X^2) method is applied. Feature selection is the the method of choosing a portion of the words found in the training corpus and Chi-Square algorithm is to reduce unnecessary words and improves the performance of the system. The final step is to classify the comments or reviews using Naive Bayes Classifier.

3.1. Chi-Square Statistics (X^2-Statistics)

Chi-Square Statistics is used to measure the independence or relevance of class variables. There are two main types of chi-square tests: chi-square goodness-of-fit test and chi-square test of independence. Chi-square goodness-of-fit test compares the observed frequencies of a categorical variable with the expected frequencies, assuming a specific distribution. In this system, it is used that the data focuses on a categorical variable (only mobile data services).

Chi-Square goodness-of-fit test is a statistical method for calculating the discrepancy between two events' anticipated and observed frequencies. The two events are the occurrence of the term and the occurrence of the class in feature selection. The value for each term t with respect to the value of class c is calculated by using word goodness measure as follows:

$$X^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C)(B+D)(A+B)(C+D)} \quad (1)$$

where, "t, c" is a term, t of a class c. The "A" is the number of documents that have the feature and belong to the specific class. The "B" is the total number of documents that do not have the particular feature but they belong to the specific class. The "C" is the number of documents that have feature and don't belong to the specific class. The "D" is the number of documents that don't have feature and don't belong to the specific class. The "N" is the total number of documents.

In this method, it needs to specify a chi-square critical value that is also known as a threshold value and compares words with chi-square values to critical value to select as features.

3.2. Naive Bayes (NB) Classifier

The supervised learning method known as the Naive Bayes, which is based on the Bayes theorem, is used to resolve classification issues. The Naive Bayes Classifier is one of the most straightforward and efficient classification algorithms available today. It aids in the development of quick machine learning models capable of making accurate forecasts. There are three main types of naïve bayes classifier: (i) Gaussian Naive Bayes: it is used for handling continuous data, where each feature is a real-valued variable, (ii) Multinomial Naive Bayes: it is used for discrete data, where each feature can take on one of several possible discrete values, and (iii) Bernoulli Naive Bayes: it is used for binary data, where each feature can take only two values (0 or 1). The proposed system is implemented by using Multinomial Naïve Bayes (MNB).

3.2.1. Multinomial Naïve Bayes (MNB)

Multinomial Naïve Bayes (MNB) classifier is a type of probabilistic algorithm used in machine learning, particularly for text classification problems. It is based on Bayes' theorem of conditional probability and assumes that the features (i.e., the words) in a text document are conditionally independent of each other, given the class label. This assumption simplifies the computation of the probability of a document belonging to a particular class.

The MNB classifier works by first calculating the prior probability of each class based on the frequency of document belonging to each class in the training data. Then, for a given document, the conditional probability of each class is calculated based on the frequency of each feature (i.e., word) in the document. The class with the highest conditional probability is chosen as the predicted class for the document.

In this system, a document D is given for classification, the probability of each class C is calculated as follows:

$$P(c \mid d) = \frac{P(c)\prod_{w \in d} P(W \mid C)^{n}wd}{P(d)} \qquad (2)$$

where, "n_{wd}" is the number of times word w occurs in document d. The "P(w|c)" is the probability of observing word w given class c. The "P(c)" is the prior probability of class c. The "P(d)" is a constant that makes the probabilities for the different classes sum to one. The "P(c)" is estimated by the proportion of training documents pertaining to class c. P(w|c) is estimated as:

$$P(w \mid c) = \frac{1 + \sum_{d \in D_c} n wd}{k + \sum_{w} \sum_{d \in D_c} n wd} \qquad (3)$$

where, "D_c" is the collection of all training documents in class c. The "k" is the size of the vocabulary (i.e., the number of distinct feature words in all training documents).

4. PROPOSED SYSTEM DESIGN

The proposed system is aimed to analyze the sentiment of comments or reviews based on machine learning techniques.

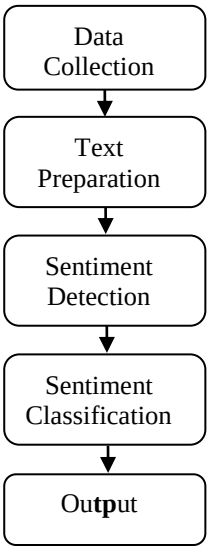


Figure 1. Overview of Sentiment Analysis Process

Figure 1 shows the overall stages of the processes involved in sentiment analysis and figure 2 shows the system flow diagram. In this system, the comments are collected on Facebook. In the training phase, preprocessing data are labeled manually for the training dataset and feature selection for selecting feature words is done. In the testing phase, an input sentence is preprocessed and feature words are chosen from the training data set.

Then classification process is carried out to define into positive, negative, and neutral.

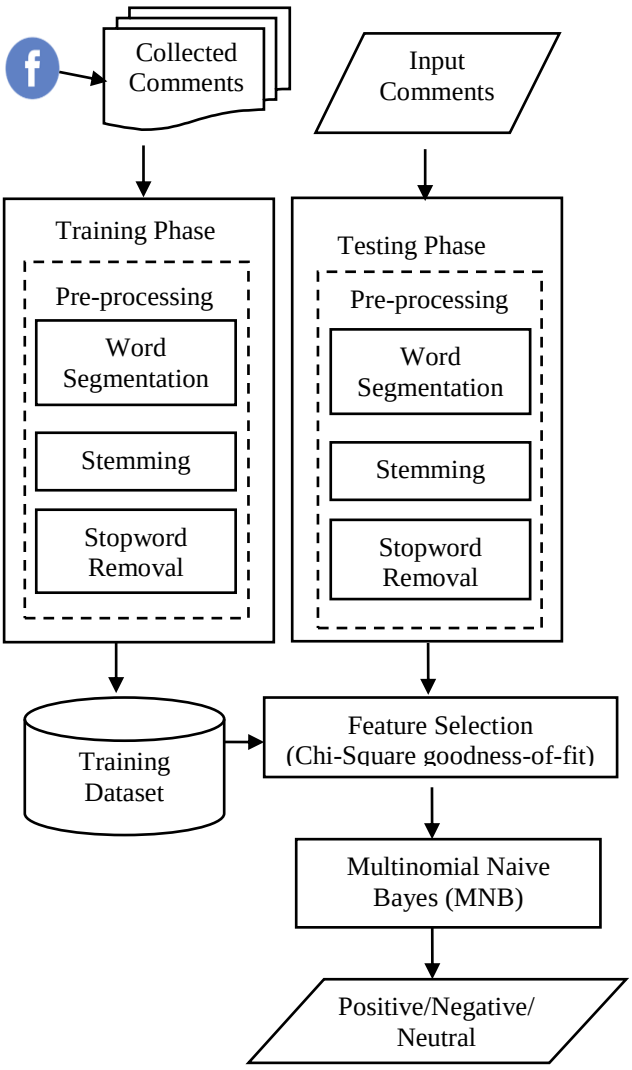


Figure 2. System Flow Diagram

4.1. Data Retrieval

The original reviews or comments from the official page of each mobile service operator on Facebook are collected using exportcomments.com. If it was the data written with Zawgyi fonts, it must be converted manually to Myanmar Standard font, Unicode.

4.2. Pre-processing of the System

In this system, pre-processing includes the word segmentation, stemming and stopwords removal processes.

4.2.1. Word Segmentation

Word segmentation is the process of segmenting sentences into valid words. This process is vital in the text classification process because the reliability of the classifier strongly depends on it.

Myanmar researchers have carried out Myanmar word segmentation in many different ways. There is no fixed rule for word segmentation in the Myanmar language. Word segmentation is important in the preprocessing stage for opinion mining processes. In this paper, (word_breaker.word_segment_v5) python library is used for Myanmar word segmentation. Table 1 shows Myanmar word segmentation Process.

Table 1. Sample Word Segmentation Process

Input	Output
MPTကမွန္မာနိုင္ငံ မ ာတာ့အကာ ဍဆုံးပါ၏	MPT, က, မွန္မာနိုင္ငံ, မ ာ,တာ့, အကာဍဆုံး, ပါ,၏
ာေဖတ္အကၠမ္ဆုံး Operator	ာေ,ဖတ္,အကၠမ္ဆုံး, Operator
ာယ္လို့ဝယ္ယူအသုံး မွ၌ ာလည္း၌ ဍမွ ပဲးပါ	ာယ္လို့,ဝယ္ယူ,အသုံး မွ,၌, ာ,လည္း,၌ ဍမွ, ပဲးပါ

4.2.2. Stemming

Stemming is a rule-based approach and the normalization technique used in Natural language processing that reduces the number of computations required. Stemming is a method of removing the prefix and suffix of the word and bringing it to a base word (e.g., "ကာေဍ" is a root-base word and "ပို့ကာေဍ", "အကာေဍဆုံး"," ကာေဍတယ္") are the different forms of a single word. So, these words get stripped out, they might get the

incorrect meanings or some other sort of errors.

4.2.3. Stopwords removal

Next step in the pre-processing is stopwords removal. Stopwords such as 'အတွက်', 'မ ', 'နဲ့', 'ရဲ့', 'ဖို့', etc. are some of the common words which occur in most of the comments. Punctuations, emoji, other unnecessary characters and typing error words (some error words may be occurred due to Unicode/Zawgyi issue) are also removed in this stage. These words are removed because they do not help to decide the class of the comments.

For detail explanation of the sentiment analysis, this system uses the sample dataset that contains a total of 15 comments and each class has 5 comments. These sample data are shown in Table 2. This dataset contains segmented words of a sentence and its corresponding label. All training documents are preprocessed which means they do not contain unnecessary characters and stopwords.

Table 2. Sample Preprocessed Comments

No.	Comments	Tag
1	ကောင်း ကြိုးစား	Positive
2	ကို်က ကို်က	Positive
3	မွန်မာနို်ငံ် ကောင်း	Positive
4	ကို်က	Positive
5	သဘောကြအားပေး	Positive
6	ဘေ မွတ်ကွမ်	Negative
…	….	…
14	ရုံး ပုန်း နံပါတ်ပွော	Neutral
15	လို်င် မကောင်း ဘယ်လို် လုပ်	Neutral

4.3. Feature Selection for Classification

After the comments are preprocessed, this system uses the feature selection method to obtain the optimal subset of features in a dataset. Feature selection is effective in the reduction of large data in text classification. This research used the Chi-Square method to reduce unnecessary words and improve the performance of the system. It is more efficient to classify by decreasing the size of vocabulary and more accurate classification by eliminating noise features.

Table 3 describes the frequencies for each term t with respect to the class c. From equation (1), Words that have a X^2 test score larger than 1.386 which indicates statistical significance at the 0.5 level are selected as features for respective classes as shown in Table 4 and will be processed in the classification stage. There are 6 feature words for positive class, 12 for negative class, and 17 for neutral class respectively.

Table 3. Frequencies for Each Term t with Respect to the Class c

t	c	A	B	C	D	N
ကောင်း	Positive	2	3	0	10	15
ကျေးဇူး	Positive	1	4	0	10	15
ကြိုက်	Positive	2	3	0	10	15
လို်င်	Negative	2	3	1	9	15
။	Neutral	1	4	2	8	15
သဘောကျ	Positive	1	4	0	10	15
မကောင်း	Negative	2	3	1	9	15
။	Neutral	1	4	2	8	15
ဖြတ်	Negative	1	4	0	10	15
ဘော	Negative	2	3	0	10	15
….	…	.	.	.	.	
ရုံး	Neutral	1	4	0	10	15

ပုနှံ	Neutral	1	4	0	10	15
ပွဓာ	Neutral	1	4	0	10	15

Table 4. Sample Selection Features

No.	Word	Class	X^2
1	ကောင်း	Positive	4.6154
2	ကျေးဇူး	Positive	2.1429
3	ကြိုက်	Positive	4.6149
4	လို့ဘူး	Negative	1.875
5	သဘောကျ	Positive	2.1429
6	မကောင်း	Negative	1.875
7	ရှုံး	Neutral	2.1429

4.4. Sentiment Analysis using MNB Classifier

Then, feature words are extracted from training dataset. It will see how many times of each feature word appears for each class. Then, the prior probability of each class is calculated to be used in classification stage. The prior probability is as follows:

- For Positive class, number of sentences of positive class in training dataset is “5”. P(Positive) = 5/15 = 1/3
- For Negative class, number of sentences of negative class in training dataset is “5”. P(Negative) = 5/15 = 1/3
- For Neutral class, number of sentences of negative class in training dataset is “5”. P(Neutral) = 5/15 = 1/3

After calculating the priori probability, this system accepts the test data. Test Data=>"MPTကမွန္မာနိုင္ငံမ ၁တစ္ေအကာင္းဆုံးပါဗ်ိ" will be classified into one of the three classes. Firstly, input sentence is segmented into words. Then, it has “MPT, က, မွန္မာနိုင္ငံ, မ ၁, တစ္ေ, အကာင္းဆုံး, ပါ, ဗ်ိ”. After stopwords are removed and the word is stemmed, two words in input sentence are included in feature words lists. These words are “မွန္မာနိုင္ငံ” and “ကာင္း”. According to equation (3), the posterior probabilities of these words for each class are as follow:

- P(မွန္မာနိုင္ငံ|Positive) =0.0488
- P(မွန္မာနိုင္ငံ|Negative) =0.0213
- P(မွန္မာနိုင္ငံ|Neutral) =0.0192
- P(ကာင္း|Positive) =0.0732
- P(ကာင္း|Negative) =0.0213
- P(ကာင္း|Neutral) =0.0192

After that, the probabilities of each class for test data are calculated using equation (2).

- P(test|Positive) = 1/3*0.0488*0.0732 = 0.00119
- P(test|Negative) = 1/3*0.0213*0.0213 = 0.0001512
- P(test|Neutral) =1/3*0.0192*0.0192 = 0.0001228

Among probability value of each class, probability value of positive class is the largest. Therefore, the input comment is classified into Positive class.

5. Implementation Of The System

The system is implemented with python language. The datasets used in the system is stored as “.csv” file format. The system is experimented on the dataset available from comments on social media network (Facebook).

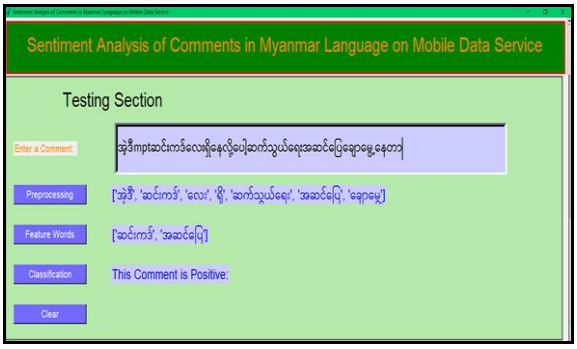


Figure 3. Testing Process about Positive Comment

Testing process about positive comment is shown in Figure 3. In the testing process for sentiment analysis, the user must first input the desired comment. Then, this system performs

the pre-processing, feature word extraction and comment classification process. After finishing classification, this system produces the output that points out the user inputted comment is positive, negative or neutral. Testing process about negative comment is shown in Figure 4.

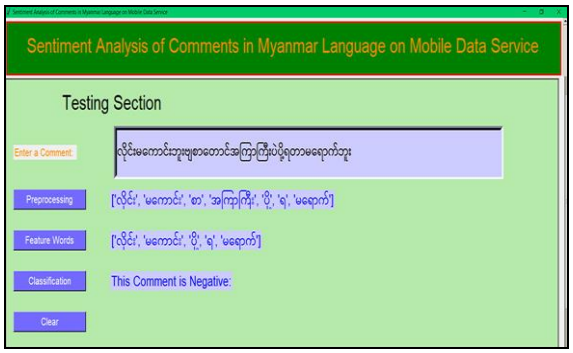


Figure 4. Testing Process about Negative Comment

6. EXPERIMENTAL RESULTS

Through several pre-processing stages, comment data is collected through official pages from Operators (ATOM, MPT, Mytel, and Ooredoo) on Facebook. In Table 5, the training data consists of over 600 comments and testing data contains 50 comments for each class.

Table 5. Testing with Each Actual Class

Class	Predicted Class	Actual Class
Positive	170	50
Negative	280	50
Neutral	150	50
Total	600	150

In this system, precision, recall and F1 score are methods used for performance measures for the test data. These are as follows:

- Precision: The percentage of comments classified as positive or negative that are actually positive or negative.
- Recall: The percentage of positive or negative comments that are correctly classified.
- F1-Score: The harmonic mean of precision and recall.

The use of chi-square method on the feature selection algorithm improves accuracy because it can reduce the words that are considered less influential for analyzing the Naïve Bayes algorithm.

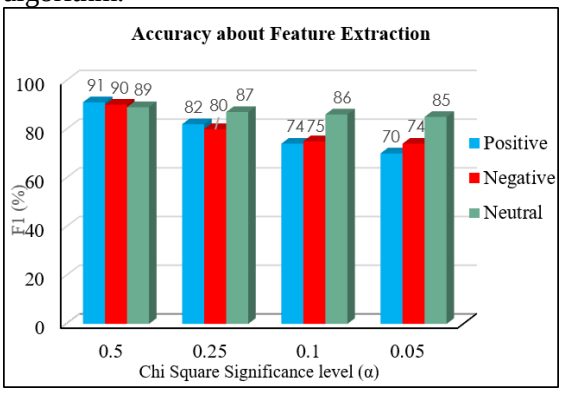


Figure 5. Comparison of Accuracy Results

The comparison of the result of the accuracy (F1%) at different Chi Square significance levels (α=0.5,0.25,0.1,0.05) for each class can be seen as shown in figure 5. The most accuracy result reached 91% of positive class, 90 % of negative class and 89% of neutral class on the significance level (α=0.5). The accuracy results of sentiment analysis can be improved based on the Chi Square significance level.

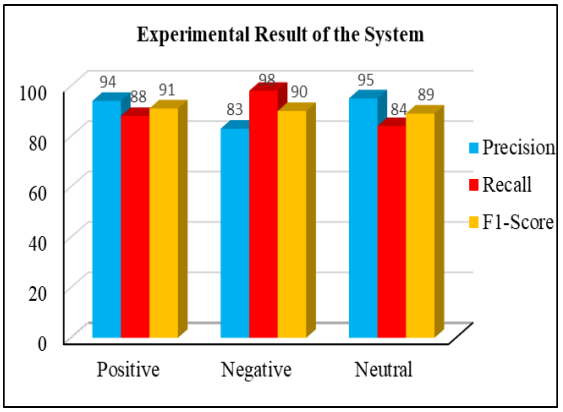


Figure 6. Graph of Result Testing System

The figure 6 shows the results of precision, recall and f1-score for categories (Negative, Neutral, Positive) using Chi Square level (α=0.5). It provides better performance of the system and increasing a result quality. The performance of the system also depends on the choice of feature selection method and number of features.

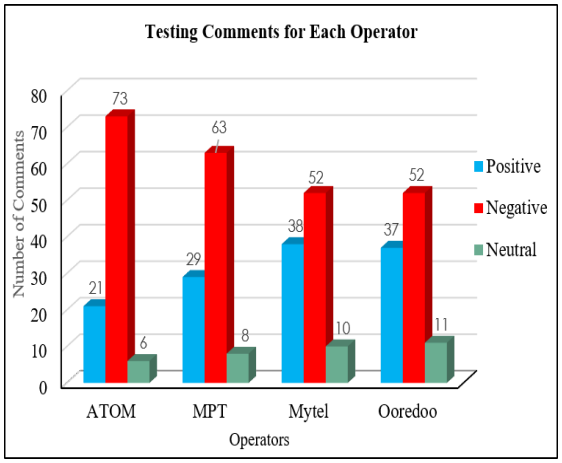


Figure 7. Testing Comments for Each Operator

In testing 100 comments from each Operator (ATOM, MPT, Mytel, Ooredoo) have several different sentiments as shown in Figure 7. Operators with the most positive number of comments are Mytel Operator of 38 comments, the most negative number of comments are ATOM Operator of 73 comments and the most neutral comments are Ooredoo Operator of 11 comments based on the training data set.

6. CONCLUSION

In this paper, Mobile Data Service Comments Dataset in Myanmar Language have been developed and the comments are classified into Positive, Negative, and Neutral from the current data set by using Multinomial Naive Bayes Classifier algorithm. And this system suggests that Mytel Operator is the one with the best service. By using the results of this system, mobile operators could know the strengths and weaknesses in improving their services. The public satisfactions about data services can be seen in this system. Therefore, this system should be used by both operators of data service providers and the public. Sentiment analysis is developing field in research for Decision Support System. The system performance testing process obtained the average precision of 91%, recall 90%, f1-score of 90% and accuracy of 90% using Chi Square level (α=0.5). For future work, we would like to compare with classifier methods such as K-NN and Support Vector Machine to get better results on the same dataset.

REFERENCES

[1] F. F. Shahare, (2017) "Sentiment Analysis for the News Data Based on the Social Media", International Conference on Intelligent Computing and Control Systems ICICCS.

[2] F. Febriyani S, Muhammad Nasrun S.Si., MT, Casi Setianingsih, ST, MT (2018), “Sentiment Analysis on the Level of Customer Satisfaction to Data Cellular Services Using the Naive Bayes Classifier Algorithm”, IEEE International Conference on Internet of Things and Intelligence System (IoTalS)

[3] H. H. Htay and K. N. Murthy (2008) "Myanmar Word Segmentation using Syllable Level Longest Matching", Proceeding of the 6th Workshop on Asian Language Resources.

[4] K. T. Nwet (2019) "Machine Learning Algorithms for Myanmar News Classification", International Journal Language Computing (IJNLC) Vol.8, No.4.

[5] Lopamudra, D., Sanjay, C., and Biswas, A. (2016) "Sentiment Analysis of Review Datasets Using Naive Bayes’ and K-NN Classifier", Information Engineering and Electronic Business, 54-62.

[6] Nurhayati, A. E. Putra, L. K. Wardhani, Busman (2019), “Chi-Square Feature Selection Effect on Naive Bayes Classifier Algorithm Performance for Sentiment Analysis Document”, The 7th International Conference on Cyber and IT Service Management, Jakarta Convention Center.

[7] R. N. Chory, M. Nasrun,C. Setianingsih (2018) “Sentiment Analysis on User Satisfaction Level of Mobile Data Services Using Support Vector Machine (SVM) Algorithm”, IEEE International Conference on Internet of Things and Intelligence System (IoTalS).

[8] S. Bahassine, A. Madani, M. A. Sarem, M. Kissi (2020), “Feature Selection using an improved Chi-square for Arabic text classification”, Journal of King Saud University - Computer and Information Sciences .

Authors

Biography

Nway Nway is a candidate for M.C.Sc, University of Computer Studies (Magway). She received B.C.Sc. degree at the University of Computer Studies (Magway), in 2018. She is a deputy staff officer working in the Treasury Department, Ministry of Planning and Finance.

Zar Lwin Phyo received her M.C.Sc. and Ph.D. (Information Technology) degrees in 2005 and 2013, respectively from University of Computer Studies, Mandalay (UCSM), Myanmar. She is a Head of the Faculty of Computer Science of University of Computer Studies (Magway). Currently she is a professor and guiding the master students. Her current research interests are in the areas of blockchain technologies, Natural Language Processing, Artificial Intelligence and Machine Learning.

EVALUATION OF MULTI VIEW TEXT CLASSIFICATION SYSTEM USING COVOLUTIONAL NEURAL NETWORKS AND LONG SHORT-TERM MEMORY

A Mar Myo Thu[1], Nandar Win Min[2], Nay Kyi Tun[3]

[1,2]University of Technology (Yatanarpon Cyber City)
[3]Myanmar Institute of Information Technology

amarmyothu2015@gmail.com, nandarwinmin@gmail.com, naykyitun.utycc@gmail.com

ABSTRACT

Analyzing the performance of multi view classification with different datasets can provide insights into the generalizability and effectiveness of multi view methods across different domains and types of text data. This paper presents an evaluation of a multi view text classification system that combines convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. The system utilizes multiple views of the input text, generated using various pre-processing techniques. The CNN component captures local patterns, while the LSTM component captures sequential dependencies. Extensive experiments on benchmark datasets demonstrate the superior performance of the proposed system compared to baseline models and traditional machine learning algorithms. The combination of CNNs, LSTMs, and multiple views offers a powerful approach for text classification.

KEYWORDS:

Multi View classification, Single-View classifiers, Deep Learning Techniques

1. INTRODUCTION

Multi-view text classification is a type of text classification that involves using multiple views or perspectives of the input text to improve classification accuracy.

In traditional text classification, features are extracted from a single view of the input text, such as the bag-of-words representation or the word embedding. However, in multi view text classification, multiple representations or views of the input text are used to capture different aspects of the text. For example, in addition to the bag-of-words representation, other views such as part-of-speech tags or sentiment analysis can be used to extract additional features.

These multiple views can then be combined in various ways to improve the accuracy of text classification. One common approach is to use a multi view learning algorithm, which learns a classifier that can effectively integrate the features from different views. Another approach is to use an ensemble of classifiers, where each classifier is trained on a different view of the input text, and the final classification is determined by combining the outputs of all the classifiers.

Multi view text classification can be useful in a variety of applications, such as sentiment analysis, topic classification, and spam filtering. By using multiple views of the input text, it can help to capture more nuanced and complex relationships between the text and the target classification, leading to more accurate and robust classification models.

Multi view text classification involves using multiple sources of information to classify text data. These sources of information can include textual, visual, and other modalities. Deep learning techniques have shown to be very effective in solving such problems.

There are a few deep learning techniques that can be used for multi view text classification.

1. Convolutional Neural Networks (CNNs): CNNs have been widely used for image

classification, but they can also be applied to text classification problems. In multi-view text classification, CNNs can be used to process the text data as well as any other modalities, such as images or audio. The output from each modality can then be combined using a fusion technique to make the final prediction.

2. Recurrent Neural Networks (RNNs): RNNs have been used extensively for sequential data processing, including text classification. In multi view text classification, RNNs can be used to process the text data and any other sequential modalities, such as audio or video. The output from each modality can then be combined using a fusion technique to make the final prediction.
3. Multi-model Learning: This involves training separate models for each modality and then combining the outputs from each model to make the final prediction. This approach has been shown to be very effective for multi view text classification.
4. Attention Mechanisms: Attention mechanisms have been used extensively in natural language processing tasks. They allow the model to focus on important parts of the input data, which can be especially useful in multi view text classification where there are multiple sources of information to consider.

2. RELATED WORKS

Multi-view text summarization is an active research area that aims to automatically generate a concise summary of a given document from multiple perspectives. In recent years, deep learning techniques have been extensively used for multi-view text summarization. Here are some related works in this area: "Multi-View Summarization with Deep Reinforcement Learning" by Wenpeng Yin et al. (2018): This work proposes a deep reinforcement learning-based approach to multi-view summarization. The model learns to generate summaries by considering multiple views of the input text.

"A Multi-View Attention-Based Neural Network Model for Text Summarization" by Xin Li et al. (2018): This work presents a multi-view attention-based neural network model for text summarization. The model uses attention mechanism to learn the importance of different views of the input text.

"Multi-View Sequence-to-Sequence Models for Summarization" by Abigail See et al. (2017): This work proposes a multi-view sequence-to-sequence model for summarization. The model uses multiple encoder-decoder pairs to generate summaries from different perspectives.

"Multi-View Convolutional Neural Networks for Text Classification" by Di You et al. (2016): This work proposes a multi-view convolutional neural network (CNN) model for text classification. The model learns multiple representations of the input text using multiple CNNs and combines them to make the final prediction.

"Multi-View Clustering for Text Summarization" by Yang Liu et al. (2015): This work proposes a multi-view clustering approach for text summarization. The model clusters the input text from multiple views and generates summaries by selecting the most representative sentences from each cluster.

These are just a few examples of the related works in multi-view text summarization using deep learning techniques. There are many other studies in this area that use different models and techniques.

3. MULTI VIEW TEXT CLASSIFICATION

The process begins with data acquisition, where the data could be collected from various sources by web crawlers or transferred from open databases. Having collected all necessary data, it is possible to verify a hypothesis that tests what type of data views (data parts) might produce the best classification model. Each data view represents a different look at the data, highlighting their peculiar properties. The process of data view selection covers two aspects, namely: (i) choosing view types, i.e. deciding which features should be included in each view and

(ii) determining the number of views. The selection of view types may be based on meta-information received from data sources: heuristic assumptions or hypotheses; prior knowledge; intuition; experience gained from experiments, etc. The number of views may be determined algorithmically or through manual data analysis.

The data views are pre-processed separately to create an independent vector space model (VSM) for each view. It is a non-trivial process consisting of the following phases. Firstly, features are constructed from data located in the views. Then, they are weighted according to a selected algorithm for features representation. As a result, we develop a number of VSMs. The number of features in the models could be reduced by features selection procedures and/ or can be projected into lower dimensional spaces. Finally, we achieve optimal VSMs representing data views.

The classification models are trained by using the VSMs and some other selected algorithms. Note that sometimes a training algorithm requires a particular VSM type. When models are properly prepared, they classify incoming data, which have to be rep-resented in the same manner as in the training phase. As a result, each classifier produces decisions regarding whether input vectors belong to each possible class. Technically, the decisions are ex- pressed as probabilities or weights. The above phases constitute the multi-view layer of the framework. The outcomes produced by the classifiers are considered as new features, and are aggregated together into a new VSM (meta-VSM). The new VSM may be projected into another feature space characterized by a lower or the same dimensionality as the one used before. The meta-VSM is a data set for training a meta-classifier. In the working mode, the meta-classifier produces the final decision, i.e. assigns a class to some given input data. The meta-classifier fuses information coming for classifiers; thus, together with the features emerging and projection processes, we call their combination the information fusion layer.

.

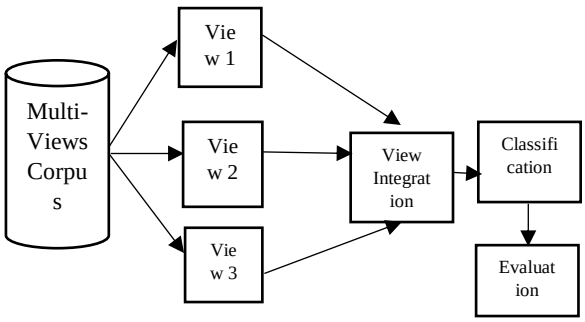


Figure 1: Flow Chart of Multi View

In this paper, we use five different datasets (AG's News, Sogou News, Yelp Review Polarity, Yahoo! Answers, Amazon Review Polarity) to classify and evaluate multi view of texts using CNN and RNN deep learning techniques.

AG's News Dataset: The AG's News dataset consists of news articles categorized into four classes: World, Sports, Business, and Science/Technology. The goal is to classify new articles into these topics. To apply the multi-view text classification system:

- Generate multiple views: Views can be created using techniques such as word embedding, character embedding, and n-grams. Each view represents a different perspective of the input text.
- Combine CNNs and LSTMs: The multi view text classification system combines CNNs and LSTMs to process each view. The CNN component captures local patterns and features, while the LSTM component captures sequential dependencies.
- Train and evaluate: The system can be trained on the AG's News dataset using labeled articles. Evaluation involves measuring metrics like accuracy, precision, recall, and F1 score on a test set. Comparison with baseline models, such as single view CNN or LSTM models, can be performed.

Sogou News Dataset: The Sogou News dataset consists of news articles in Chinese, categorized into various topics. To apply the multi-view text classification system:

- Generate multiple views: Views can be created using techniques like word embeddings, character embeddings, and n-grams, similar to the AG's News dataset.

- Combine CNNs and LSTMs: The multi-view text classification system combines CNNs and LSTMs to process each view, capturing local patterns and sequential dependencies.
- Train and evaluate: The system is trained on the Sogou News dataset using labeled articles, and evaluation metrics are calculated to assess its performance.

Yelp Review Polarity Dataset: The Yelp Review Polarity dataset consists of customer reviews from Yelp, labeled as positive or negative sentiment. The goal is to classify new reviews based on sentiment. To apply the multi-view text classification system:

- Generate multiple views: Views can be created using techniques like word embedding, character embedding, and n-grams. Combine CNNs and LSTMs: The multi-view text classification system combines CNNs and LSTMs to process each view, capturing local patterns and sequential dependencies.
- Train and evaluate: The system is trained on the Yelp Review Polarity dataset using labeled reviews, and evaluation metrics are calculated for performance assessment.

Yahoo! Datasets: The Yahoo! datasets consist of question and answer pairs, with different categories such as Sports, Society & Culture, Science & Mathematics, etc. The goal is to classify questions into the appropriate categories. To apply the multi view text classification system:

- Generate multiple views: Views can be created using techniques like TF-IDF representation, bag-of-words representation, and topic modeling (e.g., LDA) to capture different aspects of the text.
- Combine CNNs and LSTMs: The multi-view text classification system combines CNNs and LSTMs to process each view, capturing local patterns and sequential dependencies.
- Train and evaluate: The system is trained on the Yahoo! datasets using labeled question-answer pairs, and evaluation metrics are used to assess its performance.

By applying the multi-view text classification system to each of these datasets, we can leverage the benefits of combining CNNs and LSTMs with multiple views, capturing diverse information and improving classification accuracy. The specific choice of pre-processing techniques, views, and evaluation metrics may vary depending on the dataset characteristics and the specific task at hand.

4. EVALUATION

In text classification, most of open datasets are small, and large-scale datasets are splitted with a significantly smaller training set than testing. In this paper, several large-scale datasets are train for our experiments, ranging from hundreds of thousands to several millions of samples. Table 1 is a summary.

For AG's news corpus, the AG's corpus of news article are obtained on the web2.

We obtained the AG's corpus of news article on the web. It contains 496,835 categorized news articles from more than 2000 news sources. We choose the 4 largest classes from this corpus to construct our dataset, using only the title and description fields.AG News Corpus (NEWS-4) consists of news collected from up to 200 sources. All the news are divided into 4 categories based on topics. The total number of training samples is 7,600 and testing 120,000.

Sogou news corpus is a combination of the SogouCA and SogouCS news corpora, containing in total 2,909,551 news articles in various topic channels. We then labeled each piece of news using its URL, by manually classifying their domain names. This gives us a large corpus of news articles labeled with their categories. There are a large number categories but most of them contain only few articles. We choose 5 categories – "sports", "finance", "entertainment", "automobile" and "technology". The number of training samples selected for each class is 90,000 and testing 12,000.Sogou News corpus has 2,909,551 news articles and we retrieve five categories. The number of train samples in each state are 450,000 training samples and 60,000 testing samples which are used in our system. The Yelp reviews dataset is obtained from the Yelp Dataset Challenge in 2015. This dataset

contains 1,569,264 samples that have review texts. Two classification tasks are constructed from this dataset – one predicting full number of stars the user has given, and the other predicting a polarity label by considering stars 1 and 2 negative, and 3 and 4 positive. The full dataset has 130,000 training samples and 10,000 testing samples in each star, For Yelp Review Polarity, total number of review samples are 1,569,264. 560,000 training samples and 38,000 testing samples in each are used to train.

Dataset	**Classes**	**Train Samples**	**Test Sampl-es**	**Epoch Size**
AG's News	4	7,600	120,000	5,000
Sogou News	5	450,000	60,000	5,000
Yelp Review Polarity	2	560,000	38,000	5,000
Yahoo! Answers	10	1,400,000	60,000	10,000
Amazon Review Polarity	2	3,000,000	650,000	30,000

Table 1. Statistics of our large-scale datasets

We obtained Yahoo! Answers Comprehensive Questions and Answers version 1.0 dataset through the Yahoo! Webscope program. The corpus contains 4,483,032 questions and their answers. We constructed a topic classification dataset from this corpus using 10 largest main categories. Yahoo Answers dataset has total 4,483,032 questions and answers. It has 10 largest main categories and each class contains 1,400,000 trainings and 60,000 testing. The fields we used include question title, question content and best answer.

We obtained an Amazon review dataset from the Stanford Network Analysis Project (SNAP), which spans 18 years with 34,686,770 reviews from 6,643,669 users on 2,441,053 products. Similarly to the Yelp review dataset, we also constructed 2 datasets – one full score prediction and another polarity prediction. We also constructed 2 datasets – one full score prediction and another for polarity prediction. The fields used are review title and review content. The dataset contains total 3,000,000 training samples and 650,000 testing samples in each class.

For the CNN model, the dimension of word embedding d is set to 200. For the context attention, the dimension of contextual information ds is set to 100. In the convolutional layer, we use 5 filters views of text by changing the filter size respectively. Instead of concatenating different views, we apply parallel and series attention mechanisms on multi-view representation to extract the multi-view and multi-granularity attributes for enhanced performance.

For the LSTM model, we use word embedding of dimension 300, each word in the input view will be represented as a vector of length 300. Therefore, if a view consists of a sequence of N words, the input to the LSTM for that view will have a dimensionality of N x 300. In multi-view text classification, multiple views are processed separately by the LSTM, and the outputs of the LSTMs for each view are combined in some way (e.g., concatenation or averaging) to make a final prediction. The dimensionality of the combined output will depend on how the individual LSTM outputs are combined. For example, if we have two views with LSTM outputs of dimensions M and N, respectively, and we concatenate them, the resulting combined output will have a dimensionality of M + N.

To understand the results in table 2 further, we offer some empirical analysis in this section. Errors is computed by taking the difference between errors on comparison model andour character-level ConvNet model, then divided by the comparison model error.

Overall, the dimensionality of the input for an LSTM in multi view text classification depends on the dimensionality of the representation used for each view (e.g., word embedding) and the specific approach used to combine the LSTM outputs for each view.

Table 2. Testing Errors for Each Model

Model	AG	SN	YRP	YA	ARP
CNN	9.85	8.80	5.25	29.90	5.78
LSTM	13.94	4.82	5.26	29.16	6.10

6. CONCLUSION

A multi view text classification analysis using CNN and LSTM would depend on the specific experiment and dataset used. Both CNNs and LSTMs are effective in text classification tasks. CNNs are good at capturing local patterns in the text while LSTMs are better at capturing long-term dependencies. When combining the two models (CNN-LSTM) will improve classification accuracy by leveraging the strengths of both models. Multi-view learning can improve the performance of text classification models by incorporating multiple views or representations of the same text data. Preprocessing techniques such as tokenization, normalization, and stop-word removal can also improve the performance of text classification models. Choosing the right hyper parameters such as learning rate, batch size, and number of epochs is important for achieving optimal results. Overall, a multi view text classification analysis using CNN and LSTM can be an effective approach for accurately categorizing text data, but the success of the approach will depend on the specific dataset and experiment design.

6. REFERENCES

[1] Yang et al., "Multiview Convolutional Neural Networks for Multilingual and Crosslingual Text Classification", 2019

[2] Chen et al., "Multimodal Sentiment Analysis with Word-Level Fusion and Reinforcement Learning", (2019)

[3] "Multiview Deep Learning for Text Classification" by Xia et al. (2015)

[4] Xia et al., "Multi-view Recurrent Neural Networks for Intent Detection in Human-Human-Robot Conversations", 2018.

[5] Zhang et al., "Multiview Attention-Based Neural Networks for Modeling Drug-Drug Interactions", 2019

[6] Zhang et al., "Multiview Hierarchical Attention Networks for Document Classification", 2018

[7] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia, "A large-scale, multilingual knowledge base extracted from Wikipedia", Semantic Web Journal, 2014.

[8] J. McAuley and J. Leskovec, Hidden factors and hidden topics: "Understanding rating dimensions with review text" , In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM.

A Mar Myo Thu received the bachelor of computer science from University of Computer Studies, Dawei in 2006 and Master from the University of Computer Studies, Dawei in 2010. She was working as tutor in Computer University, Dawei from 2006 to since 2010. Then, she promote as assistant lecturer in University of Technology (Yatanarpon Cyber City) at 2013. Then, she was transferred to University of Computer Studies (Pyay) from 2017 to 2021. Now she is working as associate professor at University of Technology (Yatanarpon Cyber City) form 2021 to present. She is now attended in PhD-IT at University of Technology (Yatanarpon Cyber City). Her recent interest on research area includes deep Learning and machine learning.

Nandar Win Min is from Myanmar. She got Ph.D(IT) Degree from University of Technology (Yatanarpon Cyber City), Myanmar in 2014. Now, she is currently working as an Associate Professor at the Information Science department of the University of Technology (Yatanarpon Cyber City), Pyin Oo Lwin, Myanmar. The current responsibilities are multimedia developer of e-learning content development and teaching. She has 15 years teaching experience in Information Technology specialized in Programming Languages (Java, PHP & Python), Big Data Analysis, Intelligent Information Retrieval, System Analysis and Design. She is co-supervising/supervising undergrad, masters' and doctoral students of

University of Technology (Yatanarpon Cyber City).

Nay Kyi Tun is currently working as Associate Professor in the Faculty of Computer System and Technologies at University of Myanmar Institute of Information Technology (MIIT), Myandalay. I received B.C.Tech, B.C.Tech(Hons.) and M.C.Tech degrees from Computer University(Taung Ngu) and I am currently doing Ph.D research at University of Technology (Yatanarpon Cyber City).

ESTIMATING RICE YIELD IN SHAN STATE USING LINEAR REGRESSION ALGORITHM

Aye Thida Win[1]

University of Computer Studies (Kyaing Tong)
ayethidawin@ucskt.edu.mm

ABSTRACT

In recent years, agricultural production has become a major concern for many countries, including Myanmar. Rice is the staple food in Myanmar, so its production and yield are of great importance. Regression analysis is a statistical technique for investigating the relationship between variables. This paper aims to predicate the rice yield for Eastern Shan State (Keng Tung) by using a linear Regression algorithm. The dataset used for this study is from Eastern Shan State Myanmar Agriculture Department and various Eastern Shan State region locations. The results of the study showed that the linear regression algorithm could effectively predict rice yield in Shan State, Myanmar. The findings of the study can help farmers and policymakers make informed decisions regarding rice production and ensure food security in the region.

KEYWORDS

Regression analysis, linear Regression, rice yield

1. INTRODUCTION

Rice is the staple food of many Southeast Asian countries, including Myanmar and it plays a crucial role in the country's economy and food security. Shan State is the largest state in Myanmar, and rice is the primary crop grown in the region. In recent years, the agricultural sector in Myanmar has faced several challenges, including climate change, soil degradation, and low productivity. Therefore, there is a need for accurate and reliable methods to predict rice yield in Myanmar. Accurately predicting rice yield is crucial for planning crop management, assessing food security, and estimating economic growth. One of the best ways to predict unknown values is by the use of machine learning algorithms. This work intends to develop a crop yield prediction model using machine learning. There are plenty of ML algorithms that could be used, algorithms such as Linear Regression, K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), and Neural Networks can be utilized. Linear regression is a widely used algorithm for predicting continuous values, making it suitable for predicting rice yield. This paper aims to predict rice yield in Eastern Shan State, Myanmar, using agricultural data and the Linear Regression algorithm. So, we discuss Linear Regression.

2. METHODOLOGY

The proposed model is shown in the figure. 1.

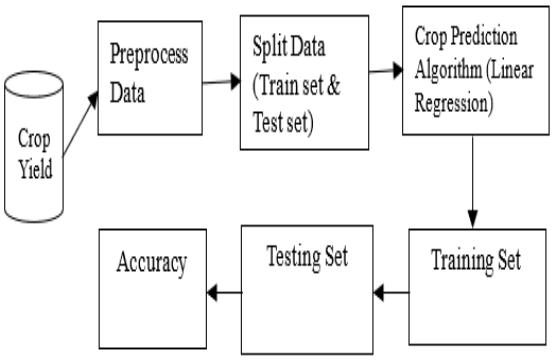


Figure 1. Linear Regression Model of the proposed system

2.1. DATASET

The dataset used for this study is from the Eastern Shan State Myanmar Agriculture Department, various Eastern Shan State region locations, and Plenty of online portals like Raitha-mithra, karnataka.gov.in, and Data.gov.in [1]. The proposed model includes a training phase and a testing phase. Four types of input data sets are used in these phases. Those are given as

follows: Location, Sown Area of Paddy (acres), Harvested Area of Paddy (acres), and Yield of Paddy (tin). The rice yield data was collected for 10 cities such as Keng Tung, Mong Khat, Mong Yang, Mong Yong, Mong Pying, Mong Tong, Mong Phyat, Tong Tar, Mong Hsat, and Mong Lar.

2.2. Linear Regression Algorithm

Linear regression is a simple and powerful machine learning algorithm that can be used to predict outcomes based on one or more predictor variables. The linear regression algorithm is a statistical method that can be used to analyze the relationship between two variables. The study uses the linear regression algorithm to predict rice yield based on the input variables. A linear regression algorithm was used to estimate crop yields in the following steps. Here's a simple algorithm step for rice yield prediction.

1. Collect data: The first step is to collect relevant data on rice yield and the predictor variables that affect it. These variables are Location, Sown Area of Paddy (acres), Harvested Area of Paddy (acres), Paddy as percentage of All Harvested, Yield of Paddy (%), and Yield of Paddy (tin). In the proposed model, we collect this data from various sources such as the Eastern Shan State Myanmar Agriculture Department, various Eastern Shan State region locations, and Plenty of online portals like Raitha-mithra, karnataka.gov.in, and Data.gov.in
2. Pre-process data: Clean the data and remove any missing values or irrelevant data. Normalize the data into a format that can be easily used by the machine learning algorithm.
3. Split data: Divide the data into training and testing sets. The training set is used to train the linear regression model, while the testing set is used to evaluate its performance. In this proposed model, using the number of datasets 20000 and a period of 5 years. Among them training dataset is 70% and the testing dataset is 30%.
4. Define the model: Now we can define the linear regression model. In its simplest form, the model can be expressed as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta n Xn \quad (1)$$

where Y is the predicted yield of rice, X_1, X_2, ..., X_n are the predictor variables, and β_0, β_1, β_2, ..., β_n are the coefficients that determine the relationship between the predictor variables and the predicted yield.

5. Train the model: We can train the model using the training set by fitting the coefficients using a method such as ordinary least squares. The method can be expressed as:

$$Y = \beta_0 + \beta_1 * X \quad (2)$$

This involves finding the values of β_0, β_1, β_2, ..., β_n that minimize the sum of squared errors between the predicted values Y and the actual values X in the training set.

6. Evaluate the model: Once the model is trained, we can evaluate its performance using the testing set. This involves comparing the predicted values with the actual values and calculating metrics such as the mean squared error or the coefficient of determination (R-squared) to assess how well the model can predict the yield of rice.
7. Make predictions: Finally, we can use the trained model to make predictions on new data. We can input values of the predictor variables for a new crop of rice, and the model will predict the corresponding yield.

3. RESULTS AND DISCUSSION

The findings of the study can help farmers and policymakers make informed decisions regarding rice production in Shan State, Myanmar. Farmers can take appropriate measures to improve their yields by understanding the factors that affect rice yield. Policymakers can use the results of the study to develop policies that support rice production and ensure food security in the region. The study also highlights the importance of using advanced statistical methods such as linear regression to predict rice yield accurately. The study found that the linear regression algorithm could effectively predict rice yield in Shan State, Myanmar. We evaluated the performance of the accuracy of a linear regression model is typically measured by metrics such as root mean square error (RMSE), mean absolute error (MAE), mean

squared error (MSE), and coefficient of determination (R2).

A. Root Mean Square Error (RMSE)

RMSE is a square root of Mean Square Error (MSE). It's the forecast errors' standard deviation. The discrepancy between the model output value and the real target value is determined first [2]. Until computing the root of the mean value, this discrepancy is squared and summed over all data elements. RMSE is expressed as in (3).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(predicted - Actual)^2}{N}} \quad (3)$$

B. Mean Absolute Error (MAE)

The average of absolute deviations between the target and predicted values are calculated as the Mean Absolute Error (MAE). The computation of MAE is shown in (4).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|predicted\ value - actual\ value| \quad (4)$$

C. Mean Squared Error (MSE)

MSE (mean squared error) or MSD (mean squared deviation) are measurements of the averages of the square error values. Its mean squared difference between both the actual and forecast values. The computation of MSE is shown in (5).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|predicted\ value - actual\ value| \quad (5)$$

D. R-Square (R2)

This is pronounced as R-squared, and this score refers to the coefficient of determination. This tells us how well the unknown samples will be predicted by our model. The best possible score is 1.0, but the score can be negative as well.

The values of MAE, MSE, and RMSE are very small, which indicates that the model is making very accurate predictions. Specifically, the MAE value of 0.005 indicates that, on average, the absolute difference between the predicted values and the actual values is very small. The MSE value of 7.67 indicates that, on average, RMSE of 0.008, R2 of 0.93 the squared difference between the predicted values and the actual values is very small. These results indicate that the predicted yield versus actual yield model predictions are close to the actual values. Therefore, in general, smaller values of MAE, MSE, and RMSE indicate better accuracy of the model.

The experiments of the proposed system results are shown below in the table and figures. The objective of the experiment was to explore the functionality of the Linear Regression algorithm and its predictive capabilities regarding yield, utilizing five parameters as input. Input data is given as follows Location: Eastern Shan State (Keng Tung), Sown Area of Paddy (acres), Harvested Area of Paddy (acres), Paddy as a percentage of All Harvested, Yield of Paddy (%), and Yield of Paddy (tin). The rice yield data was collected for 10 cities such as Keng Tung, Mong Khat, Mong Yang, Mong Yong, Mong Pying, Mong Tong, Mong Phyat, Tong Tar, Mong Hsat, and Mong Lar. The relationship between the rice planting rate and yield estimates of these cities is shown in Figure 2.

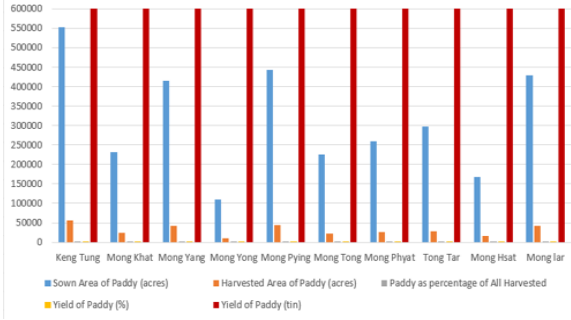


Figure 2. Rice planting rate and yield estimates of various cities

According to the System's estimate, it is estimated that the cities of Keng Tung, Mong Lar, Mong Pying, and Mong Yang may have the highest rice yield. The system predicted Keng Tung as the highest rice yield, the second highest rice yield is Mong Lar City, and followed by Mong Pying and Mong Yang rice yield are the most predicted. According to the prediction of the system, Mong Yong City is the lowest rice yield. Table 1 shows the Performance of the Applied Model.

Table 1. Performance of Applied Model

Models	Mean Square Error	Mean Absolute Error	Root Mean Square Error	R2 Score
Linear Regression	7.68	0.005	0.008	0.93

According to the use of the Linear Regression model, the results shown in Figure 3 were obtained. It is shown in the various Error Values that are MSE 7.68, MAE 0.005, RMSE 0.008 and R2 0.93. Figure 4 shows the difference between the predicted and actual values for the predicted model. And Figure 5 shows Sown Area of Paddy Acres Available and the Total Production of Paddy (tin) simultaneously for the predicted model.

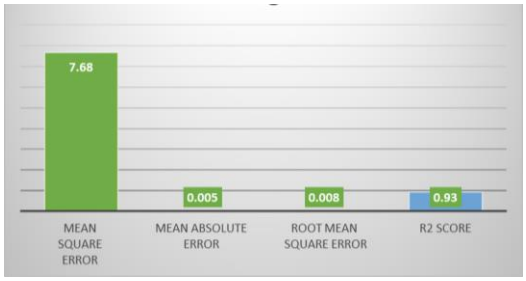


Figure 3. Various Error Values for the predicted model

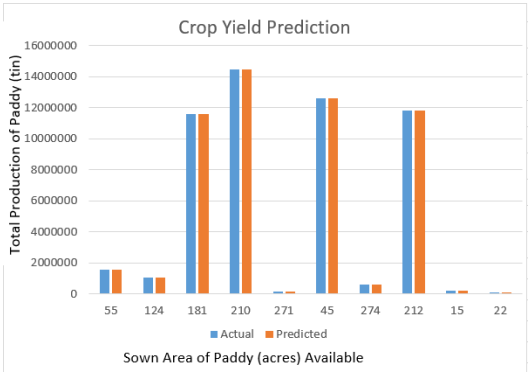


Figure 4. Difference between the predicted values and the actual values for the predicted model.

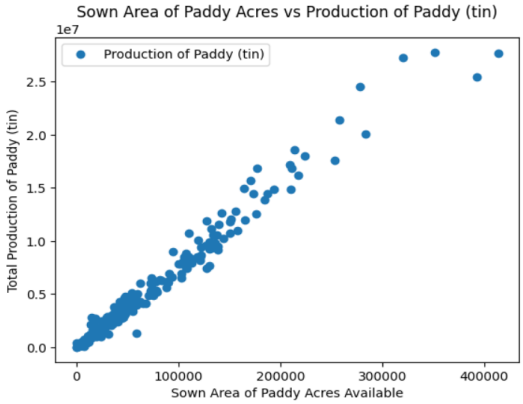


Figure 5. Sown Area of Paddy Acres and Production of Paddy (tin)

4. Conclusions

The primary goal of implementing the crop yield prediction was to acquire knowledge about crops and agriculture, while simultaneously discovering an effective method for optimizing the harvesting process. In conclusion, the study found that the linear regression algorithm could effectively predict rice yield in Shan State, Myanmar. The model achieved reasonably good accuracy, as measured by RMSE, MAE, and R2. These results suggest that the model can be useful for predicting rice yield and helping stakeholders make informed decisions about crop management and food security.

Acknowledgements

I would like to acknowledge and extend my thanks to several people who are grate interested in my paper and read it in their precious time.

References

[1] P. Vinciya “Agriculture Analysis for Next Generation High Tech Farming in Data Mining”, Anna University, Trichy, Tamilnadu, India, 5 May,2016.

[2] Safa, M., Samarasinghe, S., &Nejat, “Prediction of wheat production using artificial neural networks and investigating indirect factors affecting it: a case study in Canterbury province”, New Zealand.J. Agr. Sci. Technology, M. 2015, Vol. 17, 791-803.

[3] H. K. Karthikeya1*, K. Sudarshan2, Disha S. Shetty3, “Prediction of Agricultural Crops using KNN Algorithm”, International Journal of

Innovative Science and Research Technology, May 2020, Vol. 5, Issue. 5, ISSN No:-2456-2165.

[4] Ashwini I. Patil, Ramesh A. Medar, Vinod Desai, "Crop Yield Prediction Using Machine Learning Techniques", International Journal of Scientific Research in Science, Engineering and Technology, 09 June 2020, Vol. 7, Issue 3.

[5] Durga Prasad Sharma, Jitendra Kumar Verma, Suresh Kumar Sharma, "Study on Machine-Learning Algorithms in Crop Yield Predictions specific to Indian Agricultural Contexts", 2021 International Conference on Computational Performance Evaluation (ComPE), Dec 1-3, 2021.

[6] S Bharath, Yeshwanth S, Yashas B L, Vidyaranya R Javalagi, "Comparative Analysis of Machine Learning Algorithms in The Study of Crop and Crop yield Prediction", International Journal of Engineering Research & Technology (IJERT), Special Issue – 2020, NCETESFT - 2020 Conference Proceedings.

[7] M. A. Jayaram and Netra Marad," Fuzzy Inference Systems for Crop Yield Prediction", 2012, 21(4), pp.363-372.

[8] Veenadhari, S., Misra, B., & Singh,. "Machine learning approach for forecasting crop yield based on climatic parameters", In 2014 International Conference on Computer Communication and Informatics, C. D. (2014, January), (pp. 1-5). IEEE

[9] Niketa Gandhi, Leisa J. Armstrong, Owaiz Petkar, Amiya Kumar Tripathy, "Rice Crop Yield Prediction In India Using Support Vector Machines", 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016.

[10] Y. Dash, S. K. Mishra, and B. K. Panigrahi, "Rainfall prediction for the Kerala state of India using artificial intelligence approaches," Comput. Electr. Eng., 2018, doi: 10.1016/j.compeleceng.2018.06.004

LOCAL VISUAL FEATURE EXTRACTION AND MATCHING USING MORPHOLOGICAL OPERATION AND CONVOLUTION TECHNIQUE FOR OBJECT TRACKING

Myo Min Paing[1], Myo Min Hein[2] and Bawin Aye[3]

[1,2,3]Department of Computer Science, Higher Education Centre (Pyin Oo Lwin)
myominpaing351@gmail.com [1], myominhein48@gmail.com [2], bawinaye@ieee.org[3]

Abstract

Feature detection and matching is an important task in many applications of computer vision. A feature is a piece of information about an image's content. Local visual feature refers to small, localized regions or parts within an image that contain unique patterns of structures such as edges, corners, blobs and ridges. A corner is the intersection angle of two edges. Corner features are widely used in the fields of object tracking, image matching, image recognition, image stitching, and 3D reconstruction. In this paper, a local visual feature extraction and matching algorithm applying morphological operation and convolution technique for object tracking is proposed.

Keywords

Computer Vision, Corners, Edges, Blobs, Ridges

1. INTRODUCTION

Feature detection and matching is an important task in many applications of computer vision, such as structure-from-motion, image retrieval, object detection, object tracking and more. A feature is a piece of information about an image's content; usually describing whether a certain area of the image has certain properties. There are two types of features: global feature which represents image by one feature descriptor computed from the whole image, and local feature which represents each image by a set of local feature descriptors computed at some interesting points in the image [3]. Local visual feature refers to small, localized regions or parts within an image that contain unique patterns of structures such as edges, corners, blobs and ridges. Among them, corner features are widely used in the fields of object tracking, image matching, image recognition, image stitching, and 3D reconstruction [5]. It contains not only the information of the image, but also has the advantages of sample calculation and stable results.

There are many algorithms to detect corner features: Harris algorithm, FAST algorithm and Minimum Eigenvalue algorithm [2]. The Harris algorithm was produced by Chris Harris and Mike Stephens in 1988 by improving Moravec detector. The basic idea of the Harris algorithm is to use the central pixel point and gradient of surrounding gradient to judge the corner points, and introduce the differential operatic and autocorrelation matrix method to effectively distinguish corner points, smooth points and edge points. Proposed by Jianbo Shi and Carlo Tomasi in 1994, the Minimum Eigenvalue algorithm is improved on the algorithm of Harris with a slight modification. The FAST algorithm is used to detect specific interest points which is used to track objects in many computer vision systems, produced by Rosten and Drummond in 2006 [1].

In this research work, we proposed a local visual feature extraction and matching algorithm applying morphological operation and convolution technique. In our proposed

system, the system detects corner features of the image using edges of the image and structuring elements. And then the detected features in two frames of video are matched using convolution technique.

2. AIM

The aim of the research paper is to extract local visual features from the region of interest of the video frames using morphological operation and to match it in two consecutive frames for tracking purposes.

3. METHODOLOGY

In this paper, local visual feature extraction and matching algorithm using morphological operation and convolution technique was proposed.

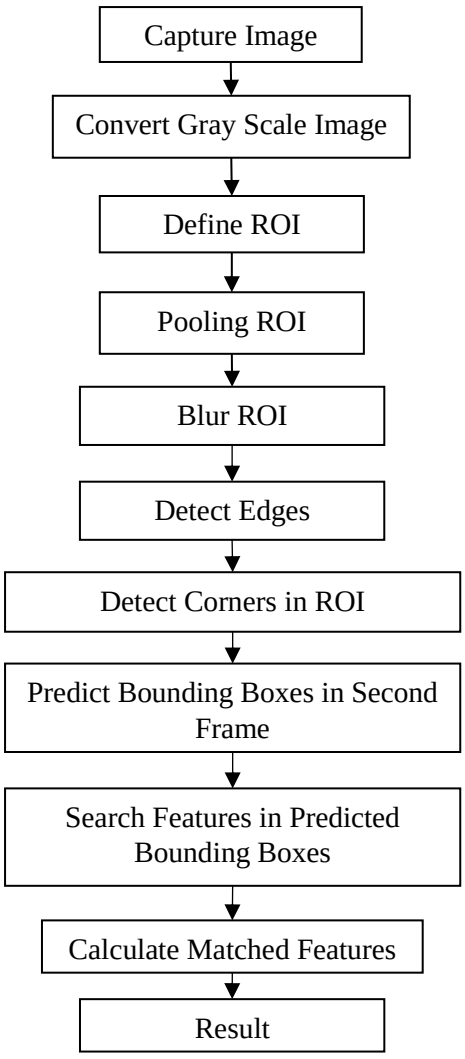


Figure 1. Flowchart of Proposed System

In the flowchart of proposed system, firstly it will capture the first frame of the video which is colour image. And then, the colour image is converted into grayscale image, and the Region of Interest (ROI) must be defined by clicking the upper left and the lower right corners of the object in the image. After that average pooling technique is applied to reduce the size of the ROI, to get specific features and for faster calculation. The result pooled image is blurred with mean filter to remove noises. Edges are detected from blurred image using Canny edge detector. Then, eight structuring elements are applied to detect corners of the image from edges. And then the second frame of the video is captured and three bounding boxes (ROIs) are predicted based on the ROI of the first frame. Detected features of the first frame are searched in predicted bounding boxes of the second frame applying convolution technique. Matched features are calculated based on Euclidean distance of the features between two frames. Finally, best fit bounding boxes are chosen based on the weight of matched features.

3.1 Mathematical Morphology

Morphology is a broad set of image processing operations that process images based on shapes. The word morphology refers to form and structure. Morphological operations use a small template known as a structuring element (SE). The structuring element is placed in all pixel locations of the image and is compared to the corresponding neighboring pixels [4]. The structuring element applied to a binary image can be described as a small matrix with a value of 0 or 1.

When a structuring element is positioned in a binary edge image, each value in the structuring element is associated with the corresponding values of the image under the structuring element. The structuring element is said to fit an image, if each value in the structuring element is equal to the corresponding pixel value of the image [4].

3.2 Corner Detection

The captured color image is converted into a grayscale image first. Then, the region of interest (ROI) must be defined manually by clicking the upper left corner and the lower right corner of ROI in which the desired object is presented.

$$ROI = I_g(R1{:}R2, C1{:}C2) \quad (1)$$

where *R1* and *R2* are the initial row and the final row of the image, C1 and C2 are the initial column and the final column of the image and I_g is the grayscale image.

ROI is interpolated using average pooling technique to resize ROI, to get specific features of ROI, and for faster calculation as shown in figure 2.

Then, the interpolated image is blurred applying mean filter to remove noises.

$$I_{blur} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} ROI(i,j) \quad (2)$$

Edges are detected from blurred image using Canny operator and the edges of the blurred image are got with the values of 0 and 1.

Then, local visual features of the image, corners, are detected using eight structuring elements as follows:

$$C_{(i,j)} = \sum_{i=1}^{m} \sum_{j=1}^{n} [I_{Edge}{}_{(i,j)} \times SE_{(i,j)}] \quad (3)$$

where C is corner response function, I_{Edge} is binary image or edges of the image and SE is structuring element.

$$F(x,y) = \begin{cases} (j,i) & if\ C == T \\ None & else \end{cases} \quad (4)$$

Where F is corner features of the image which includes x and y location of the feautres and T is Threshold. If the corner response function, $C_{(i,j)}$, is equal to threshold, in other words, $SE_{(i,j)}$ fits to $I_{Edge}{}_{(i,j)}$, we can conclude that *i* and *j* are x and y locations of corner features.

$$T = \sum_{i=1}^{m} \sum_{j=1}^{n} SE(i,j) \quad (5)$$

Where *T* is the threshold and it is the sum of the values of structuring element.

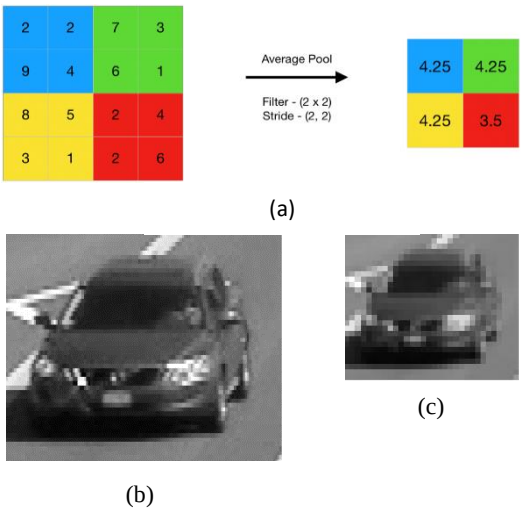


Figure 2. (a) Average Pooling, (b) Image Before Pooling, (c) Image After Pooling

Figure 2 is the illustration of average pooling technique used in our proposed system. The size of the filter is 2 × 2 and the stride of the filter is also 2 × 2.

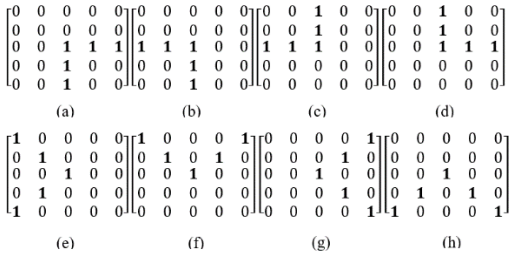


Figure 3. Eight Structuring Elements

In figure 3, eight structuring elements applied to detect corner features of the image are illustrated.

Figure 4. Corners Detected by Applying Morphological Operation

In figure 4, edges of the ROI and corners detected in ROI by applying morphological operation are illustrated. The green color points are detected corners and the white color pixels are edges of the ROI.

3.3 Feature Matching

Feature matching with convolution technique is a popular approach used in image processing for the identification and matching of specific features in images. Convolution is a mathematical operation that can be applied to an image using a predefined filter or kernel.

In our proposed algorithm, filter or kernel is defined by using the location of detected corner features, F, as follows:

$$f = I_g(F_x - i{:}\,F_x + i, F_y - j{:}\,F_y + j) \quad (6)$$

where f is a filter or kernel extracted from the grayscale image and F_x and F_y are the x and y locations of the detected features, F.

The size of the filter, f_s, can be defined as follows:

$$f_s = (i \times 2) + 1, (i = \{1,2,3, \dots .\}) \quad (7)$$

According to equation 7, if we want to get the filter with the size of 3×3, we must set one to i or if we want to get the filter with the size of 5×5, we must set two to i, etc.

Then, three possible ROIs or bounding boxes are predicted in the second frame of video as shown in figure 5.

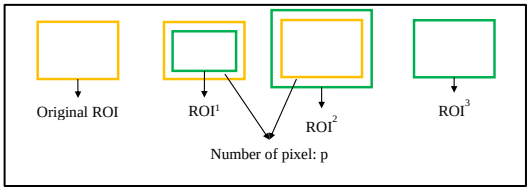


Figure 5. Original and Three Predicted ROIs

In figure 5, original ROI and three predicted ROIs are illustrated. The rectangles with the yellow outlines are original ROI and the rectangles with the green outlines are predicted ROIs. The number of pixels between original ROI and predicted ROIs is denoted as p. In ROI^3, the size of original ROI and predicted ROI is the same. So, ROI^3 can only be seen with the green outline.

After predicting three ROIs, grayscale conversion, pooling and blurring processes are applied to these ROIs. Then ROIs are convolved with filters to search the detected features of the first frame in the second frame as follows:

$$W^n = ROI^n * f \quad (8)$$

$$W_{(i,j)} = 1 - \frac{\sum_{x=-m}^{m} \sum_{y=-m}^{m} |h_{(i-x)(j-y)} - f_{xy}|}{255 \times m^2} \quad (9)$$

where W is weight matrix, h is the receptive field of predicted ROI^n, f is filter or kernel extracted based on detected features.

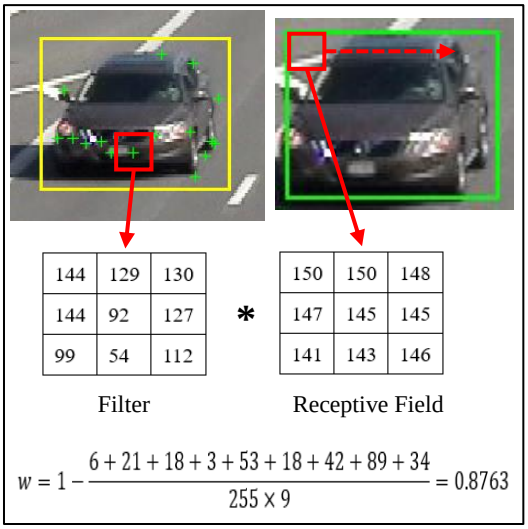


Figure 6. Calculating Weights

After calculating weight matrix, x and y locations of features found in the second frame are calculated as follows:

$$NF^n[i^*, j^*] = \max(W[i,j]), \quad (10)$$

$$1 \le i \le m, 1 \le j \le n$$

where $NF^n[i^*, j^*]$ is the x and y location of found features of ROI^n in the second frame.

The distance of features in two frames are calculated using Euclidean distance formula as follows:

$$D^n(i) = \sqrt{(F_i(x) - NF_i^n(x))^2 + (F_i(y) - NF_i^n(y))^2} \quad (11)$$

where $D^n(i)$ is the distance between i[th] features in the predefined ROI and the predicted ROI, ROI^n, in two frames.

Then matched features are calculated based on the distance of features, $D(i)$.

$$MF_i^n = \begin{cases} NF_i^n(x,y) & if\ D^n(i) \le T \\ None & else \end{cases} \quad (12)$$

Where MF_i^n is the matched features between the ROIs of two frames and the threshold, T, is defined by looking at the number of pixels between original ROI and predicted ROI.

Then the weights of three predicted ROI are calculated as follows:

$$BW^n = \frac{NMF^n}{TF} \quad (13)$$

where BW^n is the weight of bounding box, NMF^n is the number of matched features and TF is the number of detected features of the first frame.

Finally, one bounding box are chosen as best fit bounding boxes according to its weights.

4. EXPERIMENTAL RESULT

Proposed feature extraction and matching algorithm is experimented in MATLAB R2021a platform. To validate the performance of the proposed algorithm, a video file supported by MATLAB has been utilized. Comparison results of the algorithms can be seen in table 1 which compares the number of detected features and the number of matched features, using different filter sizes.

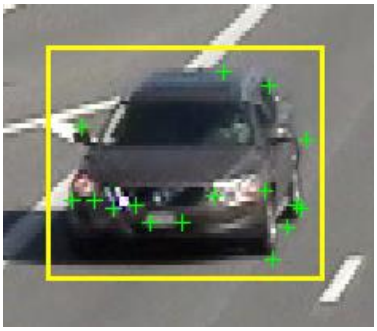


Figure 7. Detected Features in Original ROI of the First Frame

In figure 7, detected features in the original ROI of the first frame is illustrated. The green points are detected corners and the number of it is 16.

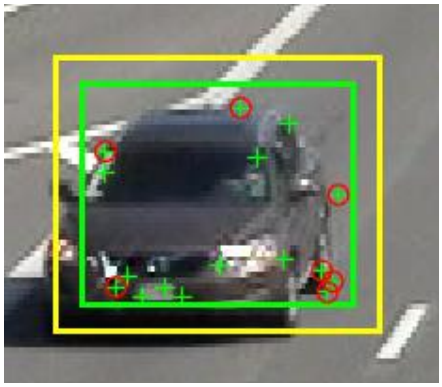


Figure 8. Matched Features in ROI[1]

In figure 8, the green points are the features found in the first predicted ROI of the second frame and the green points with the red circle are the features matched with the features of first frame. Total number of the features is 16 and total number of the matched features is 7. So, according to equation (13) the weight of the bounding box is 0.4375.

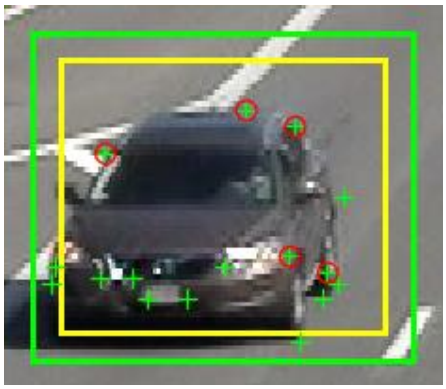


Figure 9. Matched Features in ROI[2]

In figure 9, the green points are the features found in the second predicted ROI of the second frame and the green points with the red circle are the features matched with the features of first frame. Total number of features is 16 and total number of matched features is 5. So, according to equation (13) the weight of this bounding box is 0.3125.

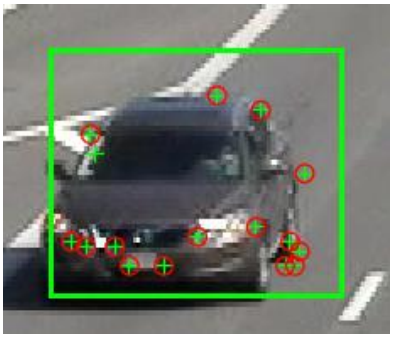


Figure 10. Matched Features in ROI[3]

In figure 10, the green points are the features found in the third predicted ROI of the second frame and the green points with the red circle are the features matched with the features of first frame. Total number of the features is 16 and total number of matched features is 15. So, according to equation (13) the weight of this bounding box is 0.9375.

Table 1. Comparison Results with Different Filter sizes

TF	f_s	*NMF*		
		ROI[1]	*ROI*[2]	*ROI*[3]
16	3x3	5	2	10
16	5x5	7	5	15
16	7x7	7	5	16

In table 1, *TF* is the number of detected features of the first frame, f_s is the size of the filter, *NMF* is the number of matched features, and *ROI*[1], *ROI*[2] and *ROI*[3] are the predicted ROIs of the second frame. In the results, most of matched features are found in ROI[3] because of not scaling too much. If the object is moving fast and the scale of the object changes very rapidly, most of the matched features can be found in ROI[1] and ROI[2].

5. CONCLUSIONS

Feature detection and matching is an important task in many applications of computer vision. In this paper, local visual feature extraction and matching algorithm using morphological operation and convolution technique is proposed. Although our proposed feature extraction algorithm gets less corner features than other algorithms, it gets more efficient and stable corner features, and less time consumption than others. Matching algorithm with predicting bounding boxes and calculating weights can track the objects in the condition of partial occlusion. So, it will be able to use in advanced military technology such as switchblade or kamikaze drone which needs to track precisely the stationary object or non-stationary object from the aerial view in the condition of partial occlusion.

ACKNOWLEDGMENT

We would like to express our sincere thanks to Dr. Soe Win Myint, Head of Department of Computer Science, and Dr. Zar Nay Linn who have, as always, given us their support throughout writing of this paper.

REFERENCES

[1] Liang Wang, Kesheng Han and Hongfei Sun, "An Adaptive Corner Detection Method Based on Deep Learning", Dept. of Flight Vehicle Engineering, Xiamen University, China, IEEE 2019.

[2] Matheel E. AbdulMunem and Eman Hato, "A Comparison of Corner Feature Detectors for Video Shot Detection", University of Technology, Bagdad - Iraq, 2018 Journal of AI-Nahrain University.

[3] Mohamed Aly, Peter Welinder, Mario Munich and Pietro Perona, "Automatic Discovery of Image Families: Global vs. Local Features", Computational Vision Lab, Caltech Pasadena, CA, USA, 16th IEEE International Conference on Image Processing (ICIP) 2009.

[4] Robert Laganie, "A Morphological Operator for Corner Detection", School of Information Technology and Engineering, University of Ottawa, Canada, Pattern Recognition Society 1998.

[5] Zhanli Li and Junchao Wang, "An Adaptive Corner Detection Algorithm Based on Edge Features", Xi'an University of Science and Technology, China, IEEE 2018.

Authors

Biography

Myo Min Paing received M.C.Sc. degree from Higher Education Center (HEC), in 2018. Now he is a Ph.D. candidate of Computer Science Department, Higher Education Center (HEC).

Dr. Myo Min Hein earned his Master's Degree from Russian State Technological University (MEPHI), (Russian Federation) in 2010. He obtained his Ph.D. (Computer Science) from HEC in 2018. Now he is an assistant lecturer of Department of Computer Science in HEC.

Dr. Bawin Aye earned his master degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2006. He also obtained his Ph.D. degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2010. Now he is a lecturer of Department of Computer Science in HEC. He is also a member of the IEEE Computer Society.

DEVELOPMENT OF MYANMAR VEHICLES' LICENSE PLATE DETECTION SYSTEM IN REAL-TIME FOR DAY AND NIGHT CONDITIONS BY USING YOLOV5 MODEL

Min Min Oo[1], Zaw Ye Aung[2], Myo Min Hein[3], Soe Win Myint[4]

[1,2,3,4]Department of Computer Technology, Higher Education Centre (Pyin Oo Lwin), Myanmar

[1]minnnminnnoo@gmail.com, [2]alexanderzaw.bmstu@gmail.com, [3]myominhein48@gmail.com,[4]soewinmyint@gmail.com

ABSTRACT

Modern technologies have made it important for citizens to feel secure in their daily lives and modes of transportation. And over the previous 20 years, the number of vehicles on the road has increased. Identification of transport vehicles has become more difficult as a result of the transportation sector's daily rapid growth. You Only Look Once (YOLO), an effective deep learning model, is used for object detection to solve that problem. Different light and weather, unavoidable data, and other challenging conditions in these conditions include acquisition noise and requirements for real-time performance in state-of-the-art (ITS) systems. This study's efficient deep learning Automatic License Plate Recognition (ALPR) method reliably is detected by training datasets of Myanmar Vehicles license plates (MVLPs). We use custom datasets collected from Yangon-Mandalay Highway. This research uses the recently released YOLOv5 object-detection framework to detect License Plates (LPs).

KEYWORDS

License Plate Detection (LPD), Automatic License Plate Recognition (ALPR), *Myanmar Vehicles License Plate (MVLP), You Only Look Once Version 5 (YOLOv5), Label Image Annotation*

1. INTRODUCTION

With the wide-ranging implementation of the intelligent transportation system in recent years. Identifying the license plate is essential in license plate recognition research before character recognition, as this contributes to the accuracy of the final recognition result. The technology for recognizing license plates in certain environments has increased accuracy in recent times. However, it is more accurate in difficult environments compared with traditional license plate detection technologies, which have limitations depending on conditions, such as dark light, hard camera distances, poor camera aspects, night headlights, and so on.

When the environment reflects the surface or color of the license plate, like in low-light conditions, the typical license plate requirement will result in failure. Videos from highway security cameras have been collected and analyzed while using the ALPR process. The frames from the videos have to be first eliminated. The second step is to identify the region of interest within the collected images. The method can be used using YOLO detection. It is a more effective develop that has real-time object detection features. After analyzing the system with a wide range of inputs to identify that the proposed method exceeds existing ones in terms of accuracy and efficiency.

2. RELATED WORK

The following results may be used in manually entering data in parking areas and toll gates on routes with a lot of vehicles. Previous research on the detection of vehicle number plates utilized a variety of techniques. With the combination of the YOLOv3 object detector and the (DNN) Deep Neural Network, Tourani

et al. presented a real-time License Plate Detection (LPD) and Character Recognition (CR) system for Iranian vehicle license plates in 2020 [1]. The networks are trained using the

realistic condition data obtained from real-world systems installed as monitoring programs, with the addition of various data augmentation techniques, so that they are robust under different conditions. On 5719 photos, the suggested method has a 95.05% average end-to-end recognition rate.

The two-stage YOLOv2 model used by Shohei et al., 2019 [2] contributed to resolving the issue with the YOLOv2 of not being able to detect smaller objects like an LP, which frequently represents just around 1% of the total area of the frame. The first stage is to identify the vehicle, while the second stage relies on detecting the license plate. While it had excellent night-time accuracy, it had difficulty detecting a distant LPs and misidentified some labels as LPs. To enable the network to adjust to the present situation.

A system with a 96.8% recognition rate for automatic license plate recognition has been proposed by Rayson [4]. The YOLO algorithm was used in this research. Due to its highly accurate Frames per Second (FPS) rates, it can identify multiple vehicle types in real-time, such as four vehicles at once in a single situation. Some number plate images may benefit significantly from the environment, including lighting, weather conditions, traffic, poor conditions, and others.

3. THE PROPOSED SYSTEM

The proposed method has been divided into two steps: The YOLOv5 model is used to detect license plates, whereas Opensource Computer Vision (OpenCV) is applied to detect video or images. The training data collection, training model, and prediction of the object by the training model are the three main components that are constantly separated when developing a machine-learning technology. Training data would be uploaded into the YOLO network to train the prediction model whenever the size of the training data exceeds the objective.

The user will finally see the results on the interface. After correctly identifying the license plate, the system removes the region which contains it and moves on to the next phase. The suggested method consists of two major parts. Those processes are testing and training. A camera is first applied to capture the input video of the highway during the training phase. The flowchart for the proposed system is shown in figure 1.

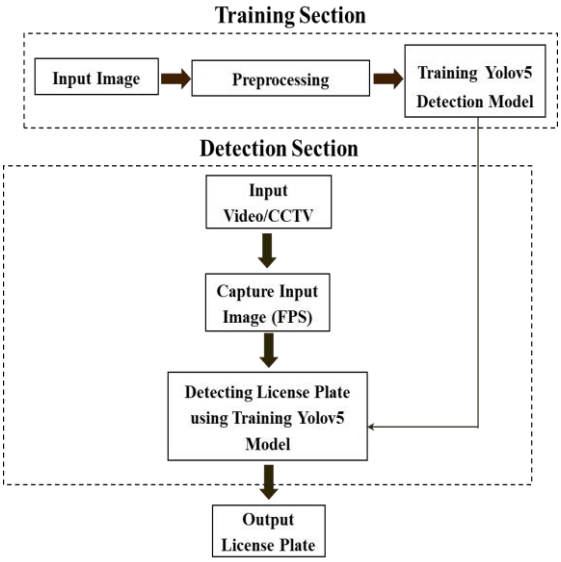


Figure 1. Proposed System Design

After that, the currently streaming video is separated into various frames, each of which is given an object classification label. Labeling is one of the steps that result from this. The annotated images will be saved and added to the dataset. The dataset will be processed through three stages: testing, validation, and training. Figure 2 shows the training images in Roboflow.

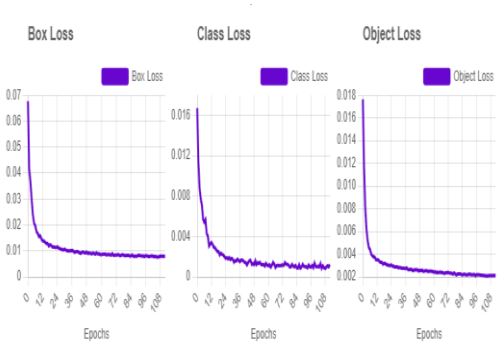


Figure 2. Training Images in Roboflow

3.1. Data Collection

The collection of data is always important for the training of the model. As a result, collecting enough data would be the first issue we would need to solve. The main objective for our research is the license plate, which is required to get. Thus, my first challenge will be working out the best way to collect the data effectively. For this experiment, training data will be almost 5,000 images and will be processed on Roboflow.com. Capturing every image is difficult, but various ways of collecting data have been proposed.

Any efforts have been made to set up smartphone cameras nearby to the Yangon-Mandalay Expressway to be able to capture video and individual data sets. PYTHON was used in this research to extract images from the video. In order to be realized, a one-second video includes either 60 or 30 frames. After extracting the image from the video, label the image using the 'LabelImg' tool that already has been integrated into the PYTHON environment. This process is known as labeling. The region where the license plate will be identified by this tool and the location should then be saved in an XML file, as illustrated in figure 3.

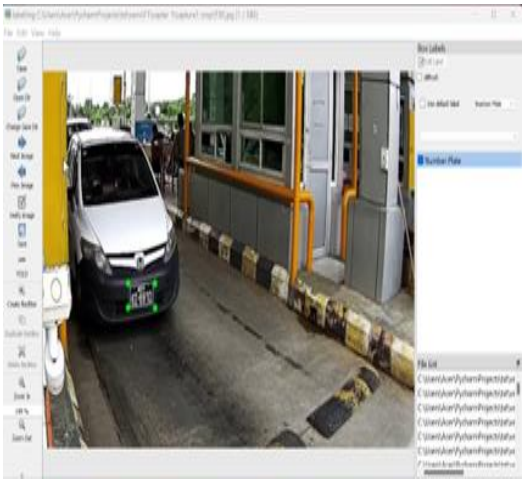


Figure 3. Labeling images using the 'LabelImg' tool

With a total percentage of 65.19%, 5000 images from self-built custom datasets were used as training datasets. A total of 1764 images for a percentage of 23.21% among validation datasets were used. The total percentage is 11% while 836 images were used in the testing datasets. Figure 4 shows the training, validation and testing images.

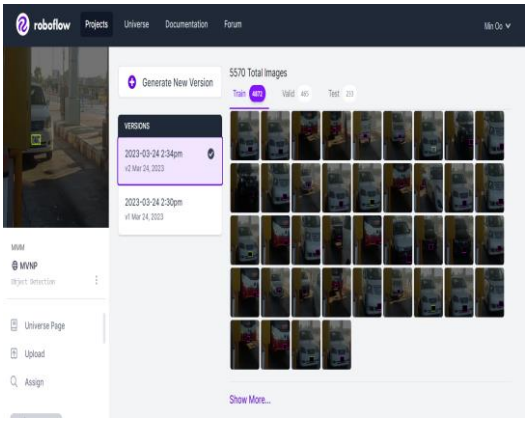


Figure 4. Training, Validation and Testing Images

3.2. License Plate Detection

The challenge's first stage is to eliminate the vehicle's license plate from an image or a video. Detecting objects typically includes two steps: 1) Object identification; and 2) object classification. Region detection is the process to detect an object in an image or video, and classification tasks, are the process of identifying an object, such as a car, bus, truck, etc. The YOLO model has been proposed to be the fastest of the various current approaches and is suitable for the real-time detection of objects. This system utilizes a single GPU due to its GPU-centric design. YOLOv5 is primarily composed of three major parts.

(a) Model Backbone

Model Backbone is mainly in the role of extracting features from images, which can be edges, shapes, lines, etc. For this, it requires a convolution neural network (CNN) with layers for batch normalization, kernel, and stride. The major system of assistance for feature extraction is the CSP (Cross Stage Partial networks). These are built on the Densenet, which connects CNN layers primarily.

(b) Model Neck

Feature pyramids are employed to identify components of an image of various sizes and to provide multi-

scale predictions. For this, feature pyramid networks are used. YOLOv5 is utilized by PaNets for FPNs (Feature Pyramid Networks).

(c) Model Head

The last detection stage, Model Head, uses an anchor to identify arrays with class probabilities, bounding box scores, and bounding boxes. The activation functions of a deep neural network are of essential use. In the middle and deep layers of YOLOv5, Leaky Relu is used, and a Nonlinear activation function is used in the top layers. The YOLOv5 model utilizes the Pytorch framework and includes 191 layers.

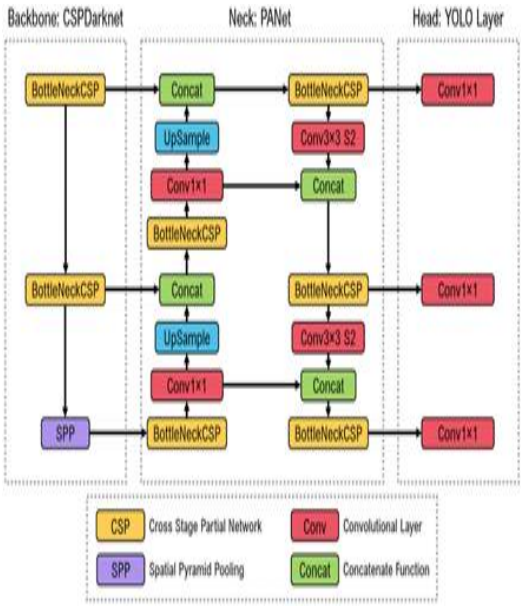


Figure 4. Network Architecture of YOLOv5

It uses K-means with neural networks to generate K-means-developed anchor boxes. The layer contains the bounding box coordinates and bounding boxes from the predictions of this network. The YOLO network will adjust the bounding box width and height to the image's width and height so that it can range from 0 to 1. This is identical to the way the other networks do it. It will be structured, which is the process of modifying the bounding box's x and y coordinates from the chosen bounding box point, to make sure that the information is also limited to the range of 0 and 1. An activation function that is linear activation of the last layer as well as all layers before it.

Figure 5. Testing image with the boundary box

4. EXPERIMENTAL RESULTS

In this research, the vehicle license plate dataset is trained and evaluated using the PyTorch deep learning framework and developed object detection-based systems for YOLOv5. The YOLOv5 model was trained on our customized dataset to detect license plates. The model uses 512-pixel images that are able to be utilized to detect and identify license plates. The majority of the application's hardware has an NVIDIA GeForce GTX 1650 graphics card, Windows 11, and the Python 3.9 programming language.

Figure 6. Experimental results of Detection in Day and Night

The results obtained from using the YOLOv5s model to detect license plates are shown in figure 7 during day and night conditions. The accuracy of actually detecting license numbers is 89%, and the false detection accuracy of not actually detecting license numbers (particularly at night) is 7%, according to

Table 1, which showed the percentage of detection accuracy rate for Myanmar vehicle number plates. The experiment's results additionally show that the accuracy of missing detection is 4%, especially at night when lights are on and visibility is poor.

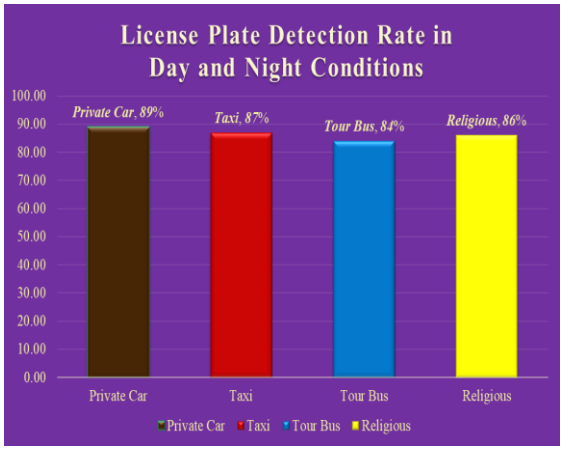


Figure 7. Experimental results of Detection Rate

Table 1. Percentage of Detection Accuracy rate in Number Plate

	True Detection NP	False Detection Np	Miss Detection NP
Total Percent of NP	89%	7%	4%

Figure 8. True Detection

Figure 9. False Detection

Figure 10. Miss Detection

3. CONCLUSIONS

This research focuses on license plate detection, allowing for real-time applications. The custom Dataset from this research was used. Our YOLOv5 model was successfully trained for object detection using the proposed method. For the detection of license plates, the accuracy is 89%. Additionally, the algorithm has been enhanced to increase the detection precision of small target license plates in road scences, which will effectively solve the problem of license plate detection in a variety of environments, both day and night.

ACKNOWLEDGEMENTS

Just would like to send a message of our appreciation to my supervisor Dr. Myo Min Hein, Assistant Lecturer at the Department of Computer Science, Higher Education Center, Dr. Zaw Ye Aung, Lecturer at the Department of Computer Science, Higher Education Center, and Prof. Dr. Soe Win Myint, Head of Department of Computer Science, Higher Education Center, for their valuable suggestions, their close supervisor, kind guidance, support, and sharing knowledge.

REFERENCES

[1] Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, and C. Y. Suen, “A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems,” IEEE Access, vol. 8, pp. 201317–201330, 2020.

[2] S. Yonetsu, Y. Iwamoto, and Y. W. Chen, “Two-Stage YOLOv2 for Accurate License-Plate Detection in Complex Scenes,” 2019 IEEE Int. Conf. Consum. Electron. ICCE 2019, no. 1, pp. 1–4, 2019, doi: 10.1109/ICCE.2019.8661944, 2019.

[3] Reza Fuad Rachmadi, and Ketut Eddy Purnamay, "Vehicle Color Recognition using Convolutional Neural Network",15 Aug, 2018.

[4] Rayson Laroca, "An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector", 2019.

[5] Rayudu Sushma,"Automatic License Plate Recognition with YOLOv5 and Easy-OCR method", IJIRT Volume 9 Issue 1, 2022.

[6] Jocher, G.; Stoken, A.; Borovec, J.; Christopher, S.T.; Laughing, L.C. Ultralytics/yolov5: v4.0-nn.SiLU() activations; Weights & Biases logging, PyTorch Hub integration. Zenodo 2021.

[7] Bradski, G. The openCV library. Dr. Dobb's J. Softw. Tools Prof. Program. 2000, 25, 120–123.

[8] Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. PLoS ONE 2021.

Authors

Biography

Min Min Oo received M.E (Computer) from Bauman Moscow State University of Technology (Kaluga) in 2011. He is currently attending Computer Technology Department, Higher Education Center (HEC) as a Ph.D. candidate. His current research includes data science; Includes computer vision and deep learning analysis.

Dr. Myo Min Hein received M.I.C.Sc Russian State Technological University (MEPHI), (Russian Federation) in 2010 and Ph.D from Defence Services Academy Computer Science Department, in 2018. He is currently working as Assistant Lecture with Computer Science Department.

Dr. Zaw Ye Aung graduated with a B.C.Sc. and was commissioned by the University of Computer Studies, Yangon (UCSY) in 2002. His Ph.D., D.Sc. (Hons), and M.E. (IT) degrees are all obtained from Bauman Moscow State Technical University. He is working as a lecturer for the Higher Education Center's Department of Computer Science.

Dr. Soe Win Myint received a master's degree from the University of Computer Studies (Yangon) in 2000 and a Ph.D. from HEC in 2017. He is currently serving as the Head of the Department of Computer Science at HEC.

IMPROVED YOLOV5 FOR REAL-TIME SMALL OBJECTS DETECTION FROM AERIAL VIEW

Kyaw Ye Aung[1], Zaw Min Khaing[2], Sai Zaw Latt[3], and Thet Naing Win[4]

[1,2,3]Department of Computer Technology, Higher Education Centre (Pyin Oo Lwin)

kyawyeaung.k52p@gmail.com [1] , zawminkhaing51@gmail.com [2] , saizawlatt46@gmail.com [3] , mr.tnwin21@gmail.com [4]

Abstract

As the advent of drone technology, object detection technology has been used to monitor illegal immigrants, industrial accidents, and natural disasters. However, since aerial images have a wider field of view and contain many small objects, it is not easy to obtain a model that is a good trade-off between accuracy and real-time performance. In order to solve this problem, in this paper, performance comparisons are analyzed between YOLOv5 models and five modified YOLOv5 models based on YOLOv5m using the VisDrone (2019) dataset, and then a model that achieves the best trade-off between accuracy and real-time performance is selected. Because aerial images typically contain only small objects, the anchor boxes of these models are recalculated using the K-means clustering method and proposed method to improve detection accuracy. Finally, it is concluded that the improved YOLOv5 achieves 80.5% mAP and 74.63 fps and is capable of meeting the requirements of accuracy and real-time performance.

Keywords

Computer Vision, Deep Learning, FPN Network, K-means, VisDrone

1. INTRODUCTION

Object detection technology on drone-captured scenarios is widely used in a variety of practical applications. Several of these applications necessitate a good trade-off between accuracy and real-time performance to detect small objects, because drone-captured images contain many small objects with a high density and confusing geographic elements.

As a result, many researchers are carrying out fascinating researches using various deep learning algorithms. Also, there are numerous deep learning algorithms, choosing the best one for real-time small objects detection in aerial images can be difficult. The YOLO versions have emerged as one of the most popular deep learning open-source frameworks. Among these versions, YOLOv5 achieves an excellent balance of accuracy and real-time performance. Some of the most significant advantages of YOLOv5 over previous versions include smaller volume, faster processing, higher precision, and implementation in the PyTorch open-source framework.

Because the objects and background are moving, real-time performance in aerial images is critical. In fact, the improved YOLOv5 based on YOLOv5m is proposed using the VisDrone (2019) dataset to achieve the best trade-off between accuracy and real-time performance.

2. AIM

The aim of this paper is to obtain a model that is the best trade-off between accuracy and real-time performance in the detection of small objects from aerial view.

3. LITERATURE REVIEWS

Deep learning detection methods are currently divided into two types: two-stage and one-stage. The sequential processing of regional proposals and classification is referred to as the two-stage method. The simultaneous processing of regional proposals and classification is referred to as the one-stage method. As a result, because real-time performance is required, most researchers use one-stage detection methods to detect small objects from aerial view.

Tasweer Ahmad et al. combined a gradient boosting classifier with YOLOv5 to detect human actions in drone images. They evaluated the modified YOLOv5 using the Okutama-Action dataset. The modified YOLOv5 achieved 75.4% mAP, but the dataset only contains human actions taken from a height of about 30 meters [1].

Xingkui Zhu et al. proposed a modified YOLOv5 based on a transformer prediction head for object detection in drone-captured scenarios. On the VisDrone (2021) dataset, they achieved 35.74% mAP by replacing the original prediction heads with transformer prediction heads. They gained nearly 7% mAP improvement and reduced layers. However, the training epochs are 65, and YOLOv5 was modified based on YOLOv5l [5].

Xue Jie et al. used an improved YOLOv5 network method for image-based ground object recognition using remote sensing. For bounding box regression, they used GIOU as the loss function. They improved YOLOv5s by changing three groups of C3 to four groups of C3, and then the head was changed to FPN+PAN. The improved YOLOv5 is tested on the DIOR dataset, which contains 20 classes. They obtained 80.5% mAP, but they trained for 200 epochs and this dataset does not include a person class [6].

Hyun-Ki Jung et al. proposed an enhanced YOLOv5 for efficient object detection using drone images in a variety of conditions. They employed the ELU activation function as well as the CIoU loss function. They improved the model's ability to detect objects in a variety of environmental and weather conditions, including clear, cloudy, rainy, snowy, day, evening, night, low and high altitudes. The enhanced YOLOv5 is tested on VisDrone and self-collected image. They received 95.5% mAP, while the original YOLOv5 received 94.6% mAP. However, their model was trained using 100 epochs and 3,360 images [3].

4. METHODOLOGY

According to the literature reviews, the YOLOv5 model achieved good mAP in small objects detection. Thus, this model is discussed in details. Following that, the modified architecture of the improved YOLOv5 model and the proposed methods for calculating the anchor box coordinates are thoroughly discussed.

4.1 YOLOv5 for Small Objects Detection

Because of its fast speed and high accuracy, the YOLO5 is one of the most well-known detection algorithms. It is the fifth version of the open-source YOLO algorithm created by Ultralytics. Glenn Jocher of UltralyticsLLC launched the YOLOv5 network on GitHub in May 27, 2020. YOLOV5 has now been upgraded to version 7. The YOLOv5 model includes ten distinct architectures: YOLOv5 (n, s, m, l, x) and YOLOv5 (s6, n6, m6, l6, x6). The abbreviations of (n, s, m, l, x) are nano, small, medium, large and extra-large. The main differences between these models are the layers and channels changed with two parameters called compound scaling – depth multiple and width multiple. The first five models have three detection scales: small, medium and large. The last five models have four detection scales: small, medium, large, and extra-large. Backbone, neck, and head are the main three components of all YOLOv5 networks. Mosaic data enhancement is used in YOLOv5 as augmentation to improve the recognition of small objects and network robustness. To extract image features, the preprocessed images are fed into the

backbone, which consists of multiple Conv, C3, and one SPPF module. The neck includes FPN and PAN. The FPN structure transports strong sematic features from the top feature maps to the lower feature maps. The PAN structure conveys strong localization features from lower to higher feature maps at the same time. These two structures work together to improve detection capability by fortifying features extracted from different network layers in backbone fusion. The head is primarily used on feature maps as a final detection step to predict objects of various size [7]. For three detection scales which use input image size (640 x 640), the default anchor box coordinates are (10,13), (16,30), (33,23), (30,61), (62,45), (59,119), (116,90), (156,198), (373,326). Similarly, for four detection scales which use input image size (1280x 1280), the coordinates are (19,27), (44,40), (38,94), (96,68), (86,152), (180,137), (140,301), (303,264), (238,542), (436,615) and (925,792). These anchor boxes are calculated using the COCO dataset, which contains various object sizes in normal view. SiLU is the activation function of YOLOv5. The CIoU loss is used to calculate the localization loss and the binary cross-entropy with logits loss is used to calculate the object and class losses. Figure 1 depicts the architecture of YOLOv5m.

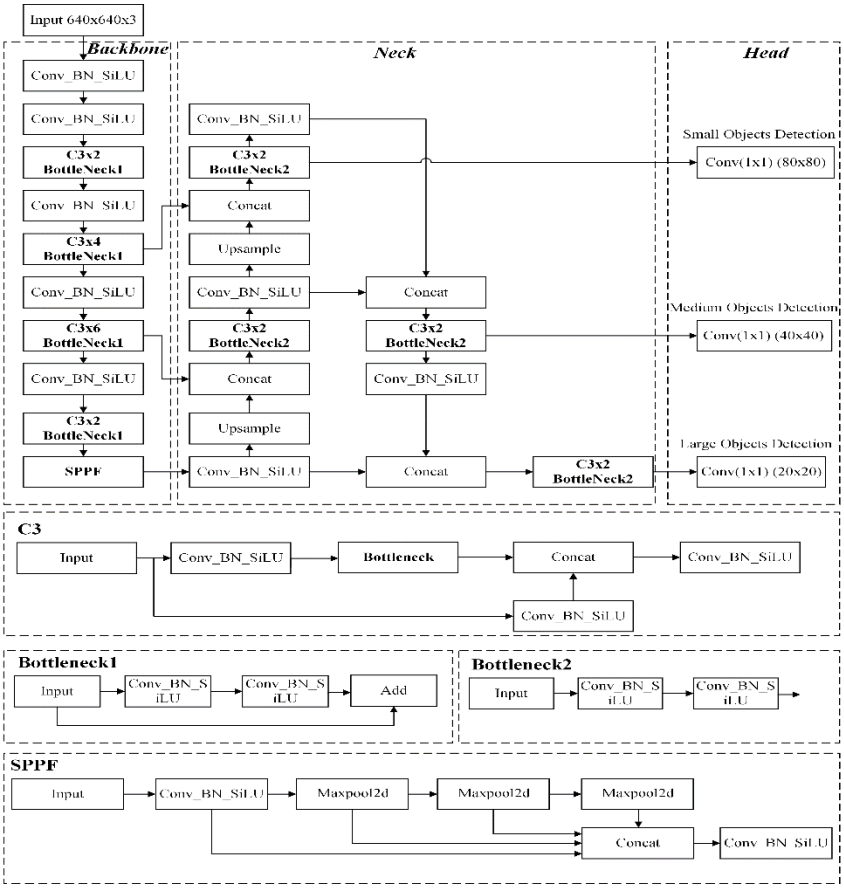


Figure 1. The Architecture of YOLOv5m

4.2 K-means Clustering for Anchor Boxes Calculation to Improve Accuracy

The original anchor boxes' coordinates of YOLOv5 models are calculated based on COCO dataset by using K-means clustering. Since the VisDrone (2019) dataset contains a large number of small and medium objects. Thus, recalculation of anchor boxes' coordinates is needed to require. K-means clustering is a partition-based clustering technique and popular for

determining the optimal number and size of anchor boxes. K-means clustering method computes distance using the Euclidean distance function [4].

$$Ed\ (x,y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\ (1)$$

Where Ed is Euclidean distance, x and y are coordinates.

In this method, larger boxes have more error clustering than smaller boxes. In fact, equation (2) is used to overcome this drawback. Furthermore, it is discovered that removing outliers from the dataset and calculating the distance improve detection accuracy. As a result, the proposed equation (3) is used to remove outliers during the pre-processing step of dataset preparation before calculating with K-means clustering.

$$d(A,B) = 1 - IoU(A,B) \qquad (2)$$

Where $d(A,B)$ is the distance between point A and point B.

$$\boldsymbol{Dataset} = (\boldsymbol{w} < \boldsymbol{\theta})\ \&\ (\boldsymbol{h} < \boldsymbol{\varphi}) \qquad (\mathbf{3})$$

Where w and h are width and height. ***θ and φ* are hyper parameters of width and height.**

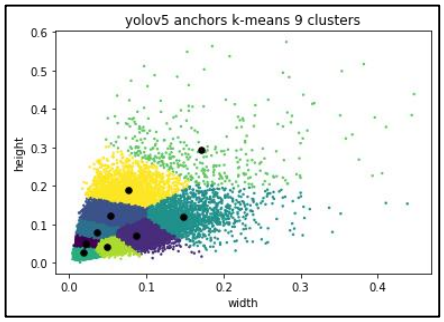


Figure 2. Data Distribution in VisDrone (2019) Dataset After Removing Outliers

In the present publication, the values of ***θ and φ*** are **0.48 and 0.58**, respectively. These values are chosen based on the experimental results.

4.3 Modified YOLOv5 for Small Objects Detection

Figure 3 depicts the architecture of modified YOLOv5. The original YOLOv5m has been altered to specialize in the VisDrone (2019) dataset.

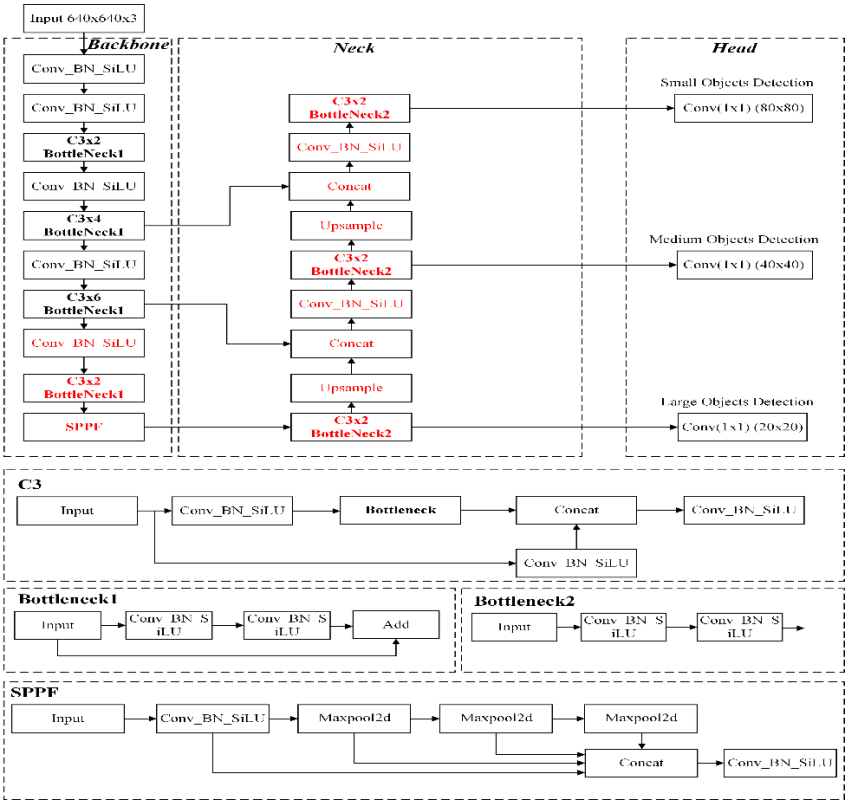


Figure 3. The Architecture of Modified YOLOv5 based on YOLOv5m

Among the YOLOv5 models, the YOLO5m model is the best model for small object detection due to its accuracy and real-time performance. The improved YOLOv5 model is proposed in this paper to reduce the complexity of the YOLOv5m model while increasing inference speed and accuracy. In figure 3, the red parts are the modifications. The improved YOLO5 has the same backbone network as the YOLO5m, but the neck has been changed by replacing the FPN network. Furthermore, while YOLOv5m uses 768 channels in the last three backbone modules, the improved YOLOv5 uses 384 channels in those modules. This is because these layers are specialized layers for large objects. Moreover, the proposed anchor box optimization method is used to improve detection accuracy and inference speed. The improved YOLOv5 has the **258** layers and **9 390 303** parameters

5. Implementation Details

The coordinates of anchor boxes for three detections are (7,14), (9,22), (12,38), (14,15), (16,27), (23,53), (25,22), (39,35) and (57,75). These coordinates are calculated by using equations (1, 2, and 3) based on VisDrone (2019) dataset. A personal computer with an Intel Core i7 (4th generation) processor (16 GB RAM) and Google Colaboratory, which supports Tesla T4 GPU (16 GB Memory) for free, are combined with the PyTorch (1.7) framework to carry out these experiments. In these experiments, the following parameters are used: batch size = 16, momentum = 0.9, weight decay = 0.0005, learning rate = 0.01, epochs = 300, and input image size = 640x640.

6. Experimental Results for Small Objects Detection

VisDrone (2019) dataset is a benchmark dataset of aerial images collected from various angles and altitudes, containing 10,219 images [2]. Detection is done in two categories: person and car. In fact, 5,500 images containing only these 2 classes were extracted in that dataset. The 5,500 images are labelled with 2 classes such as person and car using the LabelImg annotation tool. The annotated labels are saved as YOLO format. The dataset is then split up into 90% for training, 5% for validation, and 5% for testing. The experiment results of YOLOv5 (s, m and l) and five modified YOLOv5 models based on YOLOv5m are shown in table 1.

Table 1. Experimental Results for Real-Time Small Object Detection

Details / Methods	Size of Model (Megabytes)	Number of Detection Scales	Total Number of Parameters	Total Number of Layers	Frame Per Second (FPS)	mean Average Precision (mAP@0.5)
YOLOv5s	13.7 M	3	7,025,023	214	99.01 fps	76.5 %
YOLOv5m	40.2 M	3	20,875,359	291	60.98 fps	79.4 %
YOLOv5l	88.1 M	3	46,143,679	368	31.74 fps	79.6 %
YOLOv5 (proposed_1)	40.2 M	4	20,883,828	296	54.64 fps	77.8 %
YOLOv5 (proposed_2)	30.1 M	3	15,543,882	261	67.57 fps	76.6 %
YOLOv5 (proposed_3)	29.1 M	2	15,396,426	261	69.44 fps	78.5 %
YOLOv5 (proposed_4)	29.1 M	3	15,064,266	258	68.97 fps	78.9 %
YOLOv5 (proposed_5)	**18.2 M**	**3**	**9,390,303**	**258**	**74.63 fps**	**80.5 %**

7. RESULTS AND DISCUSSION

In table 1, the mean average precisions of different YOLOv5 models are investigated based on the testing dataset which have 550 images. Among the YOLOv5 (n, s, m, l and x) models, YOLOv5 (s, m, l) models have a good trade-off between accuracy and real-time performance. Thus, these models are investigated. It is found that YOLOv5s, YOLOv5m and YOLOv5l achieve 76.5% mAP, 79.4% mAP and 79.9% mAP. Moreover, these models also achieve 99.01 fps, 60.98 fps and 31.74 fps, respectively. Among these three models, YOLOv5m is the best trade-off between accuracy and real-time performance. For that reason, five modified architectures based on YOLOv5m are continued to investigate.

In the proposed_1 model, a transformer module (C3TR) module is used instead of the 8th C3 layer in the backbone. This idea is come from the literature reviews [5]. This model achieves 77.8% mAP and 54.64 fps. It is better than YOLOv5m in frames per second, but is not as good as in mAP because the features are complicated and lost. In fact, another modification is analyzed.

In the proposed_2 model, the backbone and neck are not also changed anything, but Bifpn has been added. This idea is also come from the literature reviews [6]. This model shows better frame per second than the original YOLOv5m, but not better mAP because the combination of layers and the features are complicated and lost. Thus, another improvement and modification are tried to analyze.

In the proposed_3 model, the backbone is not changed anything, but the detection scales have been changed from three to two: small and medium. Thus, there are six anchor boxes in this model. This idea is come from the discussion about detection scales in GitHub of YOLOv5. Compared to the original YOLOv5m, this model achieves higher frames per second, but it is determined to be less accurate. This is because of the combination of layers required a little bit. Therefore, another modification is also investigated.

In the proposed_4 model, the backbone is not changed anything, but the feature pyramid network (FPN) is used in the neck part. This idea is come from the architecture of YOLOv3 in order to reduce the model complexity. This model is 7.99 frames per second better than the original YOLOv5m, but 0.5% mAP less than that model. This model is the second-best model among the five modified models based on original YOLOv5m.

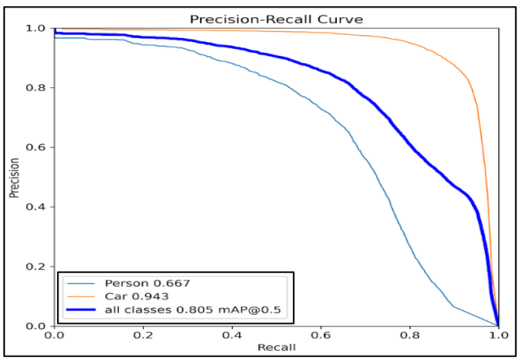


Figure 4. The Precision-Recall Curve of the Proposed_5 YOLOv5 Model

The proposed_5 model replaces the original neck with the feature pyramid network (FPN) and cuts the number of channels in half at the backbone's seventh, eighth, and ninth layers. The reason why such a channel is reduced is that those layers are specialized layers for large objects. The use of FPN in neck is to reduce the complexity of YOLOv5m. This idea is come based on the proposed_4 model. Because the detection scales are not changed, nine anchor boxes are used. According to the experimental results, this model is the best model among the five modified models. This model achieves 74.63 fps, which is 13.65 fps better than the YOLOv5m model. In addition, this model also achieves 80.5% mAP. Thus, it is 1.1% mAP better than YOLOv5m model. The Precision and Recall curve of this model is shown in figure 4.

Figure 5. The Experimental Results of Original YOLOv5m

Figure 6. The Experimental Results of Proposed Improved YOLOv5

Figure 5 and figure 6 are the experimental results using original YOLOv5 and modified YOLOv5 on aerial images. It was discovered during testing with the original YOLOv5 that some small objects could not be identified. It was discovered that the improved YOLOv5 can do detection more effectively. As a result, it is discovered that the modified YOLOv5 is superior to the original YOLOv5.

8. CONCLUSIONS

The performance comparisons between the original YOLOv5s, YOLOv5m, YOLOv5l, and updated models are examined in this paper. For investigation, the models are tested with 5,500 images containing the most people and cars in the VisDrone (2019) dataset. Since the VisDrone (2019) dataset is an aerial image dataset, the included objects are small in size, so the anchor boxes are recalculated by the proposed method. Among the five modified models, proposed_5 models based on YOLOv5m is the best model, achieving 80.5% mAP and 74.63 fps. The proposed_5 model has 24.38 fps less than YOLOv5s, but 4% mAP more than that model. Also, it is 13.65 fps and 1.1% mAP better than YOLOV5m model. Moreover, it is 42.89 fps and 0.9% mAP

better than YOLOv5l model. As a result, the proposed_5 model is a model achieving the best trade-off between accuracy and real-time performance in the detection of small objects from aerial views. In the future, the proposed_5 model's detection portion is carefully examined to increase accuracy and frame rate, the required modifications are made, and it is tested with more images.

ACKNOWLEDGMENT

I gratefully thank my supervisors Dr. Thet Naing Win, Dr. Sai Zaw Latt and Dr. Zaw Min Khaing (Lecturers at Department of Computer Technology, Higher Education Center) for valuable advices, kind guidance, supports and cooperation. Their energetic and enthusiastic supports are valuable encouragements during a period of study for this paper.

REFERENCES

[1] Ahmad, T.; Cavazza, M.; Matsuo, Y.; Prendinger, H., "Detecting Human Actions in Drone Images Using YoloV5 and Stochastic Gradient Boosting," Sensors 2022, 22,7020. doi.org/10.3390/ s22187020.

[2] D. Du et al., "VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 213-226, doi: 10.1109/ ICCVW.2019.00030.

[3] Jung, H.-K., Choi, G.-S., "Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions," Appl. Sci. 2022, 12, 7255, doi.org/10.3390/ app12147255.

[4] S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, IITSI 2010, pp. 63-67, doi: 10.1109/IITSI.2010.74.

[5] X. Zhu, S. Lyu, X. Wang and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 2778-2788, doi:10.1109/ICCVW54120.2021. 00312.

[6] Xue, J., Zheng, Y., Dong-Ye, C. *et al*, "Improved YOLOv5 network method for remote sensing image-based ground objects recognition," Soft Computer 26, 10879–10889 (2022). doi.org/10.1007/s00500-022-07106-8.

[7] Jocher, G. YOLOV5, An Open-Source Detection Model Repository. Available online:https://github.com/ultralytics/yolov5.

Authors

Biography

Kyaw Ye Aung received M.I.C.Sc. degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2016. Now he is a Ph.D. candidate of Computer Technology Department, Higher Education Center (HEC).

Dr. Zaw Min Khaing earned his Master's Degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2013. He also obtained his Ph.D. degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2018.

Dr. Sai Zaw Latt earned his master degree from Moscow Power Engineering Institute (MPEI), (Russian Federation) in 2009. He obtained his Ph.D. (Computer Technology) from HEC in 2015. Now he is a lecturer of Department of Computer Technology in HEC.

Dr. Thet Naing Win earned his master degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2007. He also obtained his Ph.D. degree from National Research University of Electronic Technology (MIET), (Russian Federation) in 2019. Now he is a lecturer of Department of Computer Technology in HEC.

CLASSIFICATION OF HEALTH CONDITION USING MACHINE LEARNING ALGORITHMS ON RESOURCE-CONSTRAINED EMBEDDED DEVICE

Ye Zay Hein[1], Thet Htoo Aung[2], and Aung Ko Ko Lwin[3]

[1,2,3]Department of Computer Technology, Higher Education Center (Pyin Oo Lwin)
yezayhein@gmail.com [1], thethtooaung@gmail.com [2], 58aungkokolwin@gmail.com [3]

ABSTRACT

Embedded technology is changing due to new developments in computer architecture and machine learning. Embedded machine learning (EML) finds application in areas like computer vision, speech recognition, healthcare, and robotics. However, implementing ML algorithms in embedded systems is challenging due to their resource-intensive nature. This study aims to create ML models on resource-constrained devices for health decision support system. To achieve this, we established a wearable sensor unit that utilized an ESP32 microcontroller to gather patient health data and location through connected temperature, pulse rate, blood pressure sensors, and a GPS module. Additionally, datasets were created for body temperature, pulse rate, blood oxygen saturation, and blood pressure. The performance of Gaussian Naive Bayes, Decision Tree, and Random Forest algorithms were evaluated. Among these, the optimized Decision Tree algorithm was selected and successfully implemented on an ESP32 development board.

KEYWORDS

Algorithms, Embedded system, ESP32, health monitoring, machine learning

1. INTRODUCTION

In recent years, the advancement of wearable technologies and the proliferation of biomedical sensors have opened new avenues for continuous monitoring of patients' health. These technologies offer an unprecedented opportunity to track vital signs and gather real-time data, empowering healthcare professionals to make informed decisions and provide timely interventions. Among the numerous applications of these advancements, the development of a continuous monitoring system that integrates location tracking with health assessment has gained significant attention.

The ability to track the location of patients in real-time is crucial in healthcare settings, as it enables healthcare providers to respond promptly to emergencies and ensure the well-being of individuals, especially in scenarios where immediate medical attention is required. Furthermore, monitoring patients' health parameters, such as temperature, heart rate, oxygen level, and blood pressure, can provide invaluable insights into their overall well-being and help detect potential diseases or health deterioration at an early stage.

To address these needs, we propose a comprehensive continuous monitoring system that combines location tracking and health assessment for patients. This system utilizes the power of biomedical sensors, such as temperature sensors, heart rate sensors, oxygen level sensors, and blood pressure sensors, which are equipped on the user's body to collect vital data. By leveraging predictive modeling techniques, the system can analyze and interpret these sensor inputs to predict potential diseases or health risks. The choice of algorithms plays

a crucial role in the accuracy and efficiency of the health decision support system.

The integration of GPS technology in the continuous monitoring system allows for real-time tracking of patients' locations. This feature ensures that healthcare providers can accurately determine the whereabouts of patients, enabling swift responses and immediate assistance when necessary. Additionally, the system incorporates a machine learning algorithm embedded in the ESP32 platform, enabling seamless communication of health status parameters to healthcare professionals or relevant stakeholders.

The objective of this article is to present the development and implementation of the proposed continuous monitoring system, highlighting its potential to revolutionize healthcare management. By combining location tracking with health assessment, our system offers a comprehensive solution for monitoring and supporting patients' well-being. The article provides an in-depth exploration of the system's architecture, the integration of predictive modeling algorithms, the utilization of biomedical sensors, and the communication of health status parameters.

2. RELATED WORK

Health Decision Support Systems (HDSS) have garnered significant attention in the healthcare field, aiming to assist healthcare professionals in making informed decisions and improving patient outcomes. In this section, we present an overview of related works on HDSS published between 2015 and 2023. These studies cover a wide range of topics, including algorithm development, data integration, decision modeling, and clinical applications, providing insights into the advancements and challenges within the field.

Ahmad, S., Baek, J., Ahn, C. R., & Kwon, G. R. examines the challenges and opportunities associated with the implementation of Health Decision Support System (HDSS). They highlight the need for effective knowledge representation, data integration, and user interface design in order to facilitate seamless decision support [1].

This study provides a comprehensive review of machine learning algorithms employed in clinical decision support systems. It evaluates the performance and applicability of various algorithms, such as Naive Bayes, Decision Trees, Support Vector Machines, and Neural Networks, in different healthcare domains [2].

Focusing on the integration of electronic health records (EHR) and decision support systems, this review explores the challenges and benefits of incorporating EHR data into HDSS. It discusses the potential for leveraging EHR data to enhance clinical decision-making and improve patient outcomes [3].

This study investigates the utilization of real-time monitoring and decision support systems for managing chronic diseases. It examines the integration of wearable devices, sensor technologies, and data analytics to provide personalized recommendations and support self-management of chronic conditions [4].

Focusing on cancer diagnosis and treatment planning, this comprehensive review highlights the advancements in clinical decision support systems. It discusses the integration of imaging data, genetic information, and clinical guidelines to assist oncologists in making accurate and timely decisions [5].

This scoping review provides an overview of artificial intelligence techniques, including machine learning, natural language processing, and expert systems, in health decision support systems. It explores their applications across different healthcare domains and identifies key research gaps and challenges [6].

Focusing on mobile health (mHealth) decision support systems, this review article discusses the applications and challenges associated with utilizing mobile technologies for decision support. It explores the potential for leveraging mobile devices, apps, and sensors to empower patients and improve healthcare delivery [7].

This review examines the role of clinical decision support systems in infectious disease management. It explores the integration of epidemiological data, diagnostic information, and treatment guidelines to support healthcare providers in the diagnosis, treatment, and prevention of infectious diseases [8].

3. ALGORITHMS

3.1 Gaussian Naive Bayes

Gaussian Naive Bayes is a variation of the Naive Bayes classifier that is used when dealing with continuous numerical features [2]. Unlike the basic Naive Bayes classifier, which assumes categorical features, Gaussian Naive Bayes can handle features that follow a Gaussian (normal) distribution. following are how Gaussian Naive Bayes works:

(a) **Assumptions:** Gaussian Naive Bayes assumes that the continuous features of each class are normally distributed. This means that the distribution of feature values for each class follows a bell-shaped curve described by the mean (average) and standard deviation of the feature values in that class.

(b) **Calculate Class Prior Probabilities:** Calculate the prior probability of each class based on the frequency of each class in the dataset. This provides a sense of how often each class appears in the data.

(c) **Estimate Mean and Variance:** For each feature in each class, calculate the mean (μ) and variance (σ^2) of the feature values. These parameters define the Gaussian distribution for each feature within a class. The mean represents the center of the distribution, and the variance determines how spread out the values are.

(d) **Calculate Gaussian Probability Density Function (PDF):** For each feature in each class, calculate the Gaussian probability density function (PDF) value of a given feature value. This PDF value represents the likelihood of observing a specific feature value given the mean and variance of the distribution. The Gaussian PDF formula is:

$$P(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i-\mu)}{2\sigma^2}\right) \quad (1)$$

Where, x is the feature value, μ is the mean, and σ is the standard deviation.

(e) **Calculate Class-Conditional Probabilities**: For a given feature vector (data point), calculate the class-conditional probability of observing its feature values given a particular class. This involves calculating the Gaussian PDF value for each feature using the corresponding class's mean and variance. Since we're assuming feature independence, the class-conditional probability for the entire feature vector is the product of the individual Gaussian PDF values for each feature.

(f) **Calculate Unnormalized Posterior Probabilities:** For each class, calculate the product of the class-conditional probabilities for all features in the feature vector, and then multiply it by the prior probability of that class. This gives you the unnormalized posterior probability for each class.

(g) **Normalize Posterior Probabilities:** Normalize the unnormalized posterior probabilities across all

classes. Divide each unnormalized posterior probability by the sum of all unnormalized posterior probabilities. This normalization step ensures that the probabilities add up to 1, making them valid probabilities.

(h) **Classification Decision:** For a given feature vector (data point), compare the normalized posterior probabilities for all classes. The class with the highest normalized posterior probability is chosen as the predicted class for that feature vector.

3.2 Decision Tree

A decision tree is a machine learning algorithm used for classification and regression tasks. It constructs a predictive model by recursively partitioning the dataset into subsets based on the values of input features [7]. The following is a step-by-step explanation of algorithm:

(a) **Select the Best Feature:** Begin by selecting the most significant feature that provides the best split in terms of information gain or impurity reduction. Information gain measures how well a feature separates classes, while impurity reduction aims to minimize the uncertainty in class assignments.

(b) **Split the Data:** Partition the dataset into subsets based on the chosen feature's values. Each subset corresponds to a branch from the current node.

(c) **Repeat for Subsets:** For each subset created, repeat the process of selecting the best feature and splitting the data. This recursion continues until a stopping criterion is met, such as reaching a maximum tree depth or having a minimum number of samples per leaf.

(d) **Assign Labels or Values:** Once the recursion stops, the leaf nodes are assigned class labels (for classification tasks) or numerical values (for regression tasks) based on the majority class or average value of samples in the leaf.

(e) **Make Predictions:** To predict the label or value for a new data point, traverse the decision tree from the root node by following the decisions based on feature values until a leaf node is reached. The prediction is the assigned class label or value of that leaf.

3.3 Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to create a robust and accurate model [4]. The following steps outline how algorithm work:

(a) **Bootstrapped Sampling:** Begin by creating multiple random subsets (bootstrapped samples) of the original dataset, allowing each subset to have different combinations of data instances.

(b) **Tree Building**: For each bootstrapped sample, build a decision tree using the process explained earlier. However, during each split, consider only a random subset of features instead of all features. This introduces randomness and diversity in the trees.

(c) **Voting or Averaging**: Once all trees are constructed, for classification tasks, each tree predicts the class of a new data point. The final prediction is determined by majority voting—selecting the class that's predicted by the majority of trees. For regression tasks, predictions from all trees are averaged to yield the final prediction.

(d) **Reducing Overfitting**: Random Forest mitigates overfitting by

averaging out the noise introduced by individual decision trees. The randomness in feature selection and the use of bootstrapped samples contribute to this effect.

Mathematically, the prediction process in Random Forest can be represented as follow:

$$y = Vote\big(T_1(x), T_2(x), \dots, T_n(x)\big) \qquad (2)$$

where y is the predicted class label, $T_i(x)$ is the prediction of the i^{th} tree for input x, and $Vote$ represents the majority voting process.

4. THE PROPOSED SYSTEM

The proposed system follows a flowchart consisting of three primary components: the sensing state, prediction state, and communication state. These components work together to create a comprehensive health decision support system and position tracking system. The flowchart begins with the system's starting point and power-on, leading to the sensing state. During this state, the system utilizes various biomedical sensors, including a body temperature sensor, heartbeat sensor, blood oxygen level sensor, and a GPS module for location tracking. These sensors collect data on the user's body temperature, heart rate, oxygen level, and current location. The collected data is then transmitted to a web server for further processing and analysis in the communication state.

In the prediction state, an embedded machine learning model, trained with datasets on body temperature and heart rate/SPO2, predicts the user's health condition based on the received sensor data. The predicted result is sent back to the web server for display. The flowchart includes a decision point to determine if the system has the capability to measure blood pressure. If a blood pressure sensor is present, the system proceeds to read the blood pressure data. This data is then utilized by an embedded machine learning model trained with a blood pressure dataset to predict the user's health condition.

The predicted result, incorporating all the vital signs, is sent back to the web server for storage and analysis, completing the communication state. This flowchart showcases the sequential steps involved in gathering vital signs, transmitting data to the web server, utilizing machine learning algorithms for prediction, and incorporating blood pressure measurements if available. The proposed system's design aims to provide a comprehensive solution for real-time health monitoring and prediction, complemented by precise location tracking, offering valuable decision-making support, and improving overall healthcare outcomes.

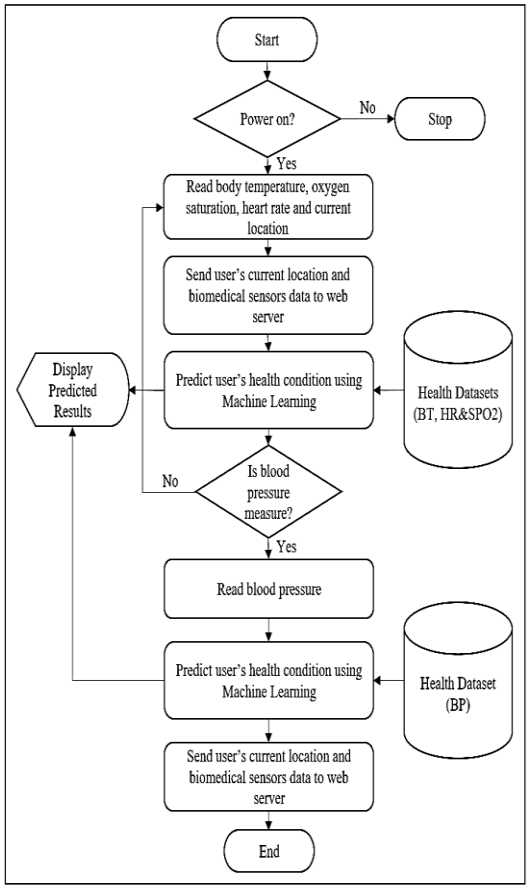


Figure 1. Proposed System Design of Health Decision Support System

4.1 Research Steps

As shown in figure 2, the research process encompassed several key steps. Firstly, datasets were meticulously constructed to capture body temperature, pulse rate, blood oxygen saturation, and blood pressure

information. Following this, data preprocessing techniques were applied to refine the collected data. Machine learning models were then trained and tested using these curated datasets. Subsequent analysis led to the selection of the optimized model. This chosen model was successfully deployed onto an embedded device, specifically an ESP32 microcontroller, which facilitated the acquisition of patient health data and location through interconnected sensors including temperature, pulse rate, blood pressure, and GPS. The predicted health status and location were displayed, while data transmission and storage were achieved through integration with a web server component.

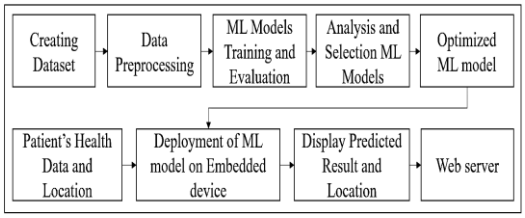


Figure 2. Process of Research

4.2 Data Collection and Pre-processing

During the data collection phase, three datasets were constructed based on globally recognized standard vital sign values, including body temperature, pulse rate, oxygen level, and blood pressure. Subsequently, in the data preprocessing stage, irrelevant data were eliminated from the collected vital sign values, and labeling was performed with the assistance of medical experts and doctors.

For the body temperature dataset, individuals' body temperatures are categorized into six medical terms (hypothermia, normal, low pyrexia, fever, high pyrexia, and hyperpyrexia) which is defined by the World Health Organization (WHO) as shown in table 1. These six classes have been further grouped into four types (healthy person, self-check, consult doctor, and emergency) to provide recommendations for seeking medical advice. The body temperature dataset consists of one feature, six classes, and a total of 600 samples.

Table 1. Body Temperature Stages

Celsius	Fahrenheit	Medical Term	Suggestion
<34	<96.8	Hypothermia	Healthy Person
34 to 37	96.8 to 98.6	Normal Range	Healthy Person
37.2 to 38.3	99 to 101	Low Pyrexia	Self Check
38.3 to 39.4	101 to 103	Fever	Consult Doctor
39.4 to 40.5	103 to 105	High Pyrexia	Emergency
>40.5	>105	Hyper Pyrexia	Emergency

Regarding the lung dataset, pulse rate and blood oxygen saturation, among the vital sign values, are used to assess an individual's lung condition. They are classified into four medical terms (bradypnea, normal, tachypnea, and hypoxia) in Table 2 based on the definitions provided by the WHO. Based on these four classes, the dataset has been divided into three types (healthy person, consult doctor, and emergency) to provide appropriate suggestions in consultation with medical professionals. The lung dataset comprises two features, four classes, and a total of 400 samples.

Table 2. Pulse Rate and Oxygen Level Stages

Systolic mmHg (Upper)	Diastolic mmHg (Lower)	Medical Term	Suggestion
<120	<80	Optimal	Healthy Person
<130	<85	Normal	Healthy Person
130-139	85-89	High Normal	Healthy Person
140-159	90-99	Mild Hypertension	Consult Doctor
160-179	100-109	Moderate Hypertension	Need Urgent
>180	>110	Severe Hypertension	Emergency
>140	<90	Isolated Systolic Hypertension	Consult Doctor

In the case of the blood pressure dataset, the systolic and diastolic vital sign values are used to determine a person's blood pressure status. In Table 3, these values are classified into seven medical terms (hypothermia, normal, low pyrexia, fever, high pyrexia, and hyperpyrexia) in accordance with the guidelines established by the WHO. Using these seven classes, the dataset has been classified into four categories (healthy person, self-check, consult doctor, and emergency) in collaboration with medical experts to provide suitable recommendations. The blood pressure dataset consists of two features, seven classes, and a total of 700 samples.

Table 3. Blood Pressure Stages

Pulse Rate (bpm)	Blood Oxygen Saturation (%)	Medical Term	Suggestion
<40	0-100	Bradypnea	Emergency
60-100	96 or more	Normal	Healthy Person
101-130	93-95	Tachypnea	Consult Doctor
131 or more	< 92	Hypoxia	Emergency

Additionally, we incorporated the Neo-6M GPS module to receive satellite data and transmit latitude and longitude coordinates. This allowed us to successfully obtain the user's location on the ESP32. The user's current location was displayed on mobile devices via Bluetooth connectivity and then sent to the server using a Wi-Fi connection.

4.3 Validation of Sensor Reading

Our proposed system employs biomedical sensors like temperature, pulse rate, and oxygen level sensors. Figure 3.4 illustrates that the sensor values obtained from our system align well with those from commercially available devices. This confirms the compatibility and suitability of these sensors for installation and use in our system.

(a) (b)

Figure 3. Validation of Sensor Value with Commercial Product

4.4 Training

During the training phase, three machine learning algorithms (Naive Bayes, Decision Tree, and Random Forest) were trained on a personal computer (PC) using carefully constructed datasets containing vital sign information. The algorithms were evaluated using metrics like accuracy, precision, recall, and F1-score to determine their performance, helping to identify the most suitable algorithm for further implementation on embedded devices. Additionally, factors such as training time, inference time, and model size were measured and analyzed to ensure efficient implementation on resource-constrained platforms.

4.5 Evaluations

The classification accuracy of the three algorithms were measured and evaluated with accuracy, precision, recall and f-score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Accuracy means the overall correctness of the classifier and the overall error rate is 1. It is defined by Eq (3).

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Precision or confidence indicates the proportion of predicted positive cases that are correctly true positives. A higher precision means less false positives, whereas a lower precision means more false positives. Precision is defined in Eq (4).

$$Recall = \frac{TP + TN}{TP + FN} \quad (5)$$

Recall or sensitivity is the proportion of true positive case that are correctly predicted positive. It measures the completeness of a classifier. Higher recall means less false negatives, while lower recall means more false negatives. It is defined by Eq (5).

$$F1 - score = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right) \quad (6)$$

F-score combines precision and recall in a harmonic mean of precision and recall calculated as Eq (6).

Where, TP represents number of true positive, FP denotes number of false positive, TN indicates number of true negative, and FN symbolizes number of false negatives.

4.6 Analysis of Machine Learning Models

Table 4 compares the performance of Naïve Bayes Classifier, Decision Tree, and Random Forest algorithms on the Body Temperature Dataset. Naïve Bayes achieves a precision, recall, and f1-score of 0.93, while Decision Tree has a precision, recall, and f1-score of 0.89, and Random Forest exhibits a precision of 0.91, with recall and f1-score of 0.9. Naïve Bayes has the highest accuracy of 0.92, followed by Decision Tree and Random Forest with accuracies of 0.88 and 0.89, respectively.

Tabel 4. Comparison of Machine learning Algorithms on Body Temperature Dataset

Dataset	Body Temperature Dataset		
Models	Naive Bayes	Decision Tree	Random Forest
Precision	0.93	0.89	0.91
Recall	0.93	0.89	0.9
f1-score	0.93	0.89	0.9
Accuracy	0.92	0.88	0.89

In Table 5, the same three classification algorithms are evaluated on the Pulse Rate and SpO2 Dataset. All models, including Naïve Bayes, Decision Tree, and Random Forest, achieve perfect precision, recall, and f1-scores of 1. Consequently, all models exhibit an accuracy of 1.

Tabel 5. Comparison of Machine learning Algorithms on Pulse Rate and SpO2 Dataset

Dataset	Pulse Rate and SpO2 Dataset		
Models	Naive Bayes	Decision Tree	Random Forest
Precision	1	1	1
Recall	1	1	1
f1-score	1	1	1
Accuracy	1	1	1

Table 6 shows how the three classification algorithms perform on the Blood Pressure Dataset. Naïve Bayes, Decision Tree, and Random Forest models achieve precision, recall, and f1-scores of 1 for Naïve Bayes, and 0.98 for both Decision Tree and Random Forest. Similarly, all models have an accuracy of 1 in this dataset.

Tabel 6. Comparison of Machine learning Algorithms on Blood Pressure Dataset

Dataset	Blood Pressure Dataset		
Models	Naive Bayes	Decision Tree	Random Forest
Precision	1	0.99	0.98
Recall	1	0.98	0.99
f1-score	1	0.98	0.98
Accuracy	1	0.98	0.98

As shown in above tables, Naïve Bayes consistently demonstrates strong performance, achieving the highest accuracy in two out of three datasets. Decision Tree and Random

Forest algorithms also exhibit competitive results with accuracies of 0.88, 0.89, and 0.98, highlighting their effectiveness in the classification tasks.

In figure 4, the model sizes were compared for three different algorithms (Naive Bayes, Decision Tree, and Random Forest) on three distinct datasets (Body temperature, Pulse Rate and SPO2, and Blood Pressure). On the Body temperature dataset, the Random Forest algorithm had the largest model size of 248,032 bytes, while Naive Bayes and Decision Tree were slightly smaller at 243,264 and 242,880 bytes, respectively. For the Pulse Rate and SPO2 dataset, Naive Bayes remained at 243,264 bytes, Decision Tree decreased slightly to 242,048 bytes, and Random Forest had a model size of 243,200 bytes. On the Blood Pressure dataset, Naive Bayes had a model size of 243,520 bytes, Decision Tree decreased to 242,208 bytes, and Random Forest increased to 244,768 bytes. These comparisons highlight the variations in model sizes across algorithms and datasets, providing insights into their memory requirements and computational efficiency.

The models yielded the following outcomes.

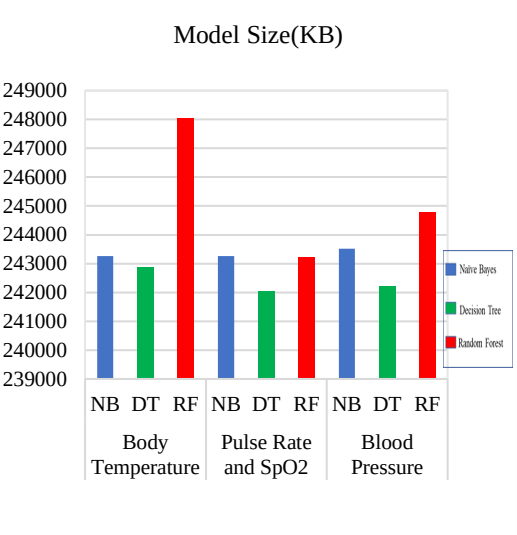


Figure 4. Machine Learning Model Size with Vital Sign Datasets

Figure 5 illustrates the inference time of each machine learning model on the embedded system with different datasets. In the body temperature dataset, Naive Bayes achieved a prediction time of 1.77 milliseconds, followed by Decision Tree with 1.468 milliseconds, and Random Forest with 1.516 milliseconds. Similarly, in the pulse rate and SpO2 dataset, Naive Bayes exhibited a prediction time of 1.740 milliseconds, while Decision Tree and Random Forest achieved 1.516 milliseconds and 1.468 milliseconds, respectively. In the blood pressure dataset, Naive Bayes achieved a prediction time of 2.043 milliseconds, whereas both Decision Tree and Random Forest models achieved a prediction time of 1.468 milliseconds. Notably, the Decision Tree model showcased the lowest prediction time across all datasets.

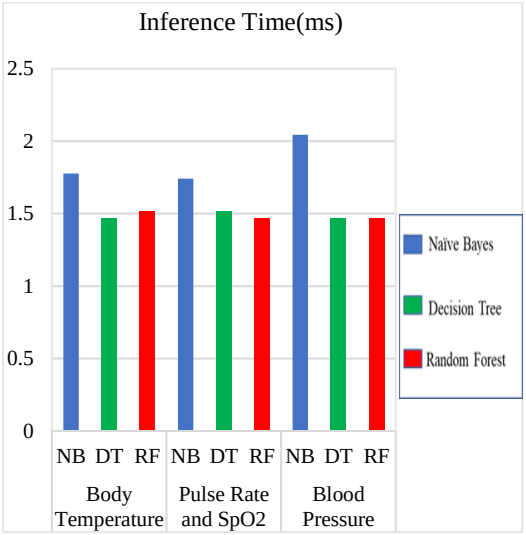


Figure 5. Inference Time of Machine Learning Models on ESP32

Figure 6 demonstrates the flash utilization percentage of each dataset on the ESP32 Development Board. The Decision Tree model exhibited the smallest flash percentage, indicating efficient memory usage. On the other hand, the Random Forest model utilized the largest flash percentage. Considering these findings, the Decision Tree model was selected as the optimized machine learning model for our embedded system due to its favorable combination of model size, flash utilization, and prediction time.

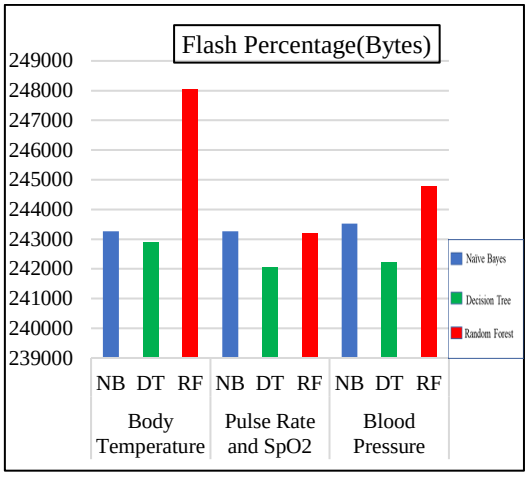


Figure 6. Flash Percentage of Each Dataset on ESP32

4.7 Deployment of Machine Learning Model on ESP32

The Decision Tree algorithm emerged as the best choice for deploying the machine learning model on the ESP32 microcontroller following a thorough evaluation. Compared to other algorithms like Naïve Bayes and Random Forest, Decision Tree consistently performed well across diverse datasets, with precision, recall, and f1-score values of 0.89 and an accuracy of 0.88 on the Body Temperature Dataset, highlighting its effectiveness. Its low inference time on the ESP32, as depicted in Figure 5, reinforced its suitability for real-time applications. Furthermore, Figure 6 indicated that the Decision Tree model used the least flash on the ESP32 Development Board, emphasizing efficient memory usage. Considering these qualities, the Decision Tree model was the optimal choice, offering a compact size, efficient flash utilization, and rapid prediction time.

As shown in figure 7, the wearable equipment is equipped with various sensors such as a temperature sensor, pulse rate sensor, blood pressure sensor, and a GPS module, all connected to an ESP32 microcontroller worn by the person. The collected data from these sensors is then processed by a machine learning model, specifically the Decision Tree algorithm, running on the ESP32. This model can predict potential health outcomes based on the sensor data. These predictions are then transmitted to the cloud server, where the data is stored and visualized in real-time. The system provides a view of the individual's health status, enabling timely interventions and healthcare management. This integration of wearable sensors, machine learning, and cloud technology holds immense potential for personalized health monitoring and early detection of health issues.

Figure 7. Integrated Wearable Health Monitoring System

4.8 User Interface

The user interface of our system generates GPS position and the user's health data, which is obtained through real-time vital sign values captured by our built model. These values are then used to generate predicted results, indicating the user's health status. The predicted results, such as "healthy person," "consult doctor," or "emergency," are compared against the predefined labels for similarity. Figure 2 and 3 visually represents the user's health data derived from sensor values, while the health decision support system utilizes the Decision Tree algorithm to predict health status. Additionally, the user's health conditions are conveniently displayed on their phone as shown in figure 8, enabling easy access and monitoring.

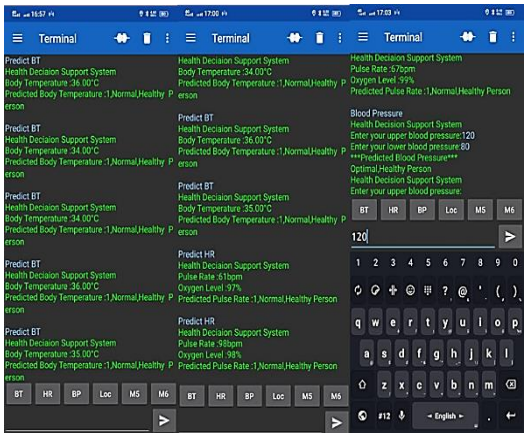


Figure 8. User's Health Prediction on Android

The user interface supports real-time monitoring through a secure Wi-Fi connection with ThingSpeak Cloud as shown in figure 9, ensuring continuous data transmission, as well as offline monitoring using Bluetooth connectivity with the user's phone devices. This comprehensive interface empowers users with valuable insights and aids in making informed decisions about their health.

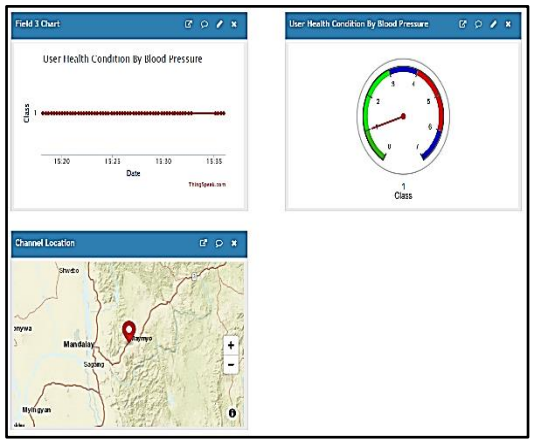


Figure 9. Health Prediction and Position Tracking on ThingSpeak Cloud Server

5. CONCLUSION

This study highlights how advancements in computer architecture and machine learning are changing embedded technology. The application of Embedded Machine Learning (EML) in fields like computer vision, speech recognition, healthcare, and robotics is promising. However, integrating resource-intensive Machine Learning (ML) algorithms into embedded systems poses challenges. The research aimed to develop ML models for health decision support system on constrained devices. By using an ESP32 microcontroller, health data and location were gathered using sensors. After creating datasets and testing three algorithms, the Decision Tree algorithm was chosen for implementation on the ESP32 board. This research showcases the potential of merging sensors and ML for enhanced healthcare monitoring.

ACKNOWLEDGEMENTS

I would like to extend heartfelt gratitude to my supervisor, Dr. Thet Htoo Aung, Lecturer at the Department of Computer Technology, Higher Education Centre, Pyin Oo Lwin, for guiding into uncharted realms of knowledge. Special thanks are also due to the co-supervisors, Dr. Yin Thway and Dr. Hlaing Moe Than, from the Higher Education Centre, Pyin Oo Lwin, for their unwavering support and invaluable advice throughout the research journey. Furthermore, I wish to express sincere appreciation to all the compassionate individuals who provided direct and indirect assistance during this research endeavor. Their contributions have been instrumental in the success of this work.

REFERENCES

[1] Ahmad, S., Baek, J., Ahn, C. R., & Kwon, G. R. (2016) "A systematic review of clinical decision support systems for therapeutic drug monitoring: characteristics, applications, and quality assessment", *Journal of Clinical Pharmacy and Therapeutics*, Vol. 41, No. 3, pp239-248.

[2] Cho, I., & Kim, D. (2016) "Decision support system for early diagnosis of Alzheimer's disease using classification algorithms", *International Journal of Distributed Sensor Networks*, Vol. 12, No. 10, pp1-7.

[3] Osheroff, J. A., Teich, J. M., Middleton, B., Steen, E. B., & Wright, A. (2017) "Clinical decision support: helping electronic health record systems", *Health Affairs*, Vol. 36, No. 12, pp1049-1057.

[4] Huang, Y., Jiang, X., & Duan, H. (2018) "Predictive analytics with ensemble learning for healthcare decision support. Expert Systems with Applications", Vol. 94, pp133-141.

[5] Souza, R. R., de Azevedo-Marques, P. M., Santos, M. A., & Filho, A. B. (2019) "Machine learning models for early detection and prediction of dengue fever in a clinical setting", *BMC Medical Informatics and Decision Making*, Vol. 19, No. 1, pp54-60.

[6] Zhang, J., Chen, J., & Zhou, M. (2020) "A cloud-based clinical decision support system for drug-drug interaction prediction", *Computers in Biology and Medicine*, Vol. 116, pp1035-1052.

[7] Li, M., Lv, Y., Zhang, H., Han, J., Wang, C., Li, Y., & Zhou, W. (2021) "An intelligent clinical decision support system for predicting mortality risk in COVID-19 patients", *Journal of Biomedical Informatics*, Vol. 116, pp1037-41.

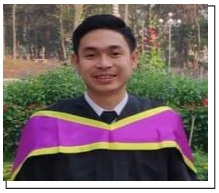
[8] Gaur, M., & Chouhan, S. (2022) "Design and evaluation of a clinical decision support system for personalized management of chronic diseases", *Health Informatics Journal*, Vol. 28, No. 1, pp3046-3061.

Authors

Biography

First A. Author Ye Zay Hein received M.C.Tech degree in Computer Technology from Higher Education Center (HEC), in 2018. He is now attending the Ph.D. 18th Batch at the Higher Education Center (HEC), Pyin Oo Lwin. Department of Computer Technology, Higher Education Center (HEC). His current research includes IoT, Embedded System and data analysis.

Second B. Dr. Thet Htoo Aung received the M.I.C.Sc. degree from the Moscow Institute of Electronic Technology (MIET), Russian Federation in 2009. He continued his studies and got his Ph.D. (Computer Technology) in 2016. He is now serving as an assistant lecturer at the Department of Computer Technology at the Higher Education Center (HEC), Pyin Oo Lwin.



Third C. Author Mr. Aung Ko Ko Lwin, a graduate of B.C.Tech, was commissioned from the Defence Services Academy in 2016. He earned his M.C.Tech in computer technology from Higher Education Center (HEC), Pyin Oo Lwin, in 2022

THE ANALYSIS AND DESIGN COMPUTERIZED INVENTORY MANAGEMENT SYSTEM FOR SUPERMARKET USING DDBMS METHODS

Nay Bala[1], Zaw Ye Aung[2], and Htin Kyaw[3]

[1,2,3] Department of Computer Science, Higher Education Centre, Pyin Oo Lwin
naybala53@gmail.com, alexanderzaw.bmstu@gmail.com,
mgnyein48@gmail.com

ABSTRACT

Effective inventory management is crucial for the success of organizations in today's fast-paced business environment. This paper explores the design and implementation of a computerized inventory management system using Distributed DBMS methods. It is designed and implemented using ASP.NET and Oracle database. It examines the challenges faced, proposed solutions, and benefits of utilizing Distributed DBMS in inventory management. This study investigates real-world case studies and existing research to highlight the advantages of scalability, fault tolerance, real-time data access, and enhanced data integrity offered by Distributed DBMS. This paper contributes to the growing body of knowledge in the field of inventory management. It uses Distributed DBMS methods, guiding organizations in their pursuit of effective inventory management solutions in today's dynamic business landscape.

KEYWORDS

Inventory management system, Distributed data base management system, ASP.NET, Oracle DB.

1. INTRODUCTION

In today's fast-paced business environment, effective inventory management plays a crucial role in the success of organizations. Every process was based on paperwork, human fault rate was high, the process and the tracing the inventory losses were not possible, and there was no efficient logging systems [1]. The ability to efficiently track, monitor, and control inventory levels is vital for optimizing operational efficiency, reducing costs, and meeting customer demands. The task of inventory management is to find the quantity of inventories that will fulfil the demand, avoiding overstocks [2]. Inventory management is the process of effectively controlling the continuous movement of items into and out of a stock of products that already exists. Traditional inventory management systems often struggle to handle the increasing complexity and scale of modern businesses. The main objective of the proposed system is to facilitate the overall operation of the organization. To overcome these challenges, the design and implementation of a computerized inventory management system using Distributed Database Management System (DDBMS) methods have gained significant attention.

A DDBMS is a database management system that stores data across multiple interconnected computers or nodes, allowing for increased scalability, fault tolerance, and improved performance. By utilizing Distributed DBMS methods, businesses can enhance their inventory management systems by achieving real-time data access, efficient data synchronization, and improved data integrity.

The design and implementation of a computerized inventory management

system using DDBMS methods involves various considerations and challenges. Firstly, the system must handle the dynamic nature of inventory data, including stock levels, supplier information, and customer orders, and ensure accurate and up-to-date information across multiple locations. Additionally, the system needs to support concurrent access and data consistency, allowing multiple users to access and update inventory information simultaneously while maintaining data integrity.

Moreover, scalability is a critical aspect when dealing with large-scale inventory management. As businesses expand and deal with increased volumes of data, the system should be capable of accommodating the growing demands and efficiently distributing the data across multiple nodes.

Furthermore, security and data privacy are of utmost importance in inventory management systems, especially when dealing with sensitive information such as pricing, customer details, and trade secrets. The design and implementation of Distributed DBMS methods should incorporate robust security measures to protect the inventory data from unauthorized access or breaches.

The database design is crucial for the reason that it forms the basis of the supermarket management system. A distributed database system involves storing data across multiple physical locations, and Oracle offers features that address the challenges of distributed data management [3]. Oracle Database's combination of scalability, data replication, consistency, high availability, fault tolerance, security, performance optimization, centralized management, and support make it a choice for managing distributed databases.

This paper aims to explore the design and implementation of a computerized inventory management system using distributed DBMS methods. It will investigate the various challenges and considerations involved in designing such a system and propose potential solutions and best practices. The paper will also analyse real-world case studies and existing research to highlight the benefits and limitations of using Distributed DBMS methods in inventory management systems.

By examining the design and implementation of computerized inventory management systems using distributed DBMS methods, this research contributes to the growing body of knowledge in the field of inventory management and provides insights for organizations seeking to enhance their inventory management capabilities.

2. RELATED WORK

Several studies and research papers have explored the design and implementation of computerized inventory management systems using Distributed DBMS methods. These works have provided valuable insights into the challenges faced, proposed solutions, and the benefits of utilizing Distributed DBMS in inventory management.

Another research paper by Abisoye Opeyemi A. presented a case study of a computerized inventory management system in the manufacturing industry. The authors emphasized the importance of data consistency and concurrent access control in managing inventory across different stages of the production process. Their test was be carried out on their system after an installation of Visual Basic 6.0 with Microsoft Access package installation in the system. [4]

Monica Kadam et.al presented cloud service using cloud DBMS methods. Cloud database management system is a distributed database that delivers computing as a service. It is sharing of web infrastructure for resources, software and information over a network. [5]

Liling Xia presented a distributed inventory management system based on intranet

system structure. He analysed and designed the function model of distributed inventory management system, and introduced system design and implementation methods, and implement the valid management and fast and accurate retrieval of distributed inventory information. [6]

Yongchang Ren presented design and implementation of supermarket management system. He designed and implemented the model of supermarket management system by using using Java language and Oracle as the background database. [7]

These related works demonstrate the diverse applications and advantages of utilizing distributed DBMS methods in the design and implementation of computerized inventory management systems. They provide valuable insights into the challenges faced in managing distributed inventory data, propose effective solutions, and highlight the benefits of scalability, fault tolerance, real-time data access, data integrity, and security. By building upon the findings of these studies, this research aims to contribute to the existing knowledge and provide further understanding of the design and implementation of computerized inventory management systems using distributed DBMS methods.

3. RESEARCH METHODOLOGY

In this article, Distributed Database management system (DDBMS) methods are mainly utilized. The function structure diagram as shown in Fig (1).

Inventory management is the system, which use to order, distribute, store, sale of goods and move inventory through the supply chain. It ensures we have the right amount of product in the right place at the right time. The goal of inventory management is to minimize the cost of holding inventory, while keeping stock levels consistent and getting products into customers' hands faster. Inventory management is the heart of a successful retail business.

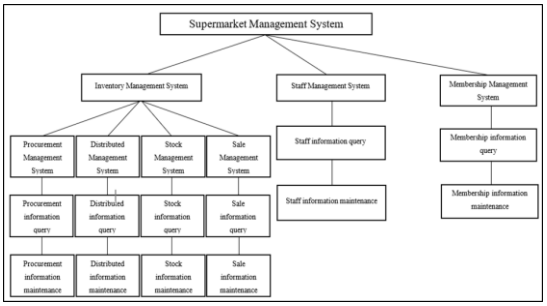


Figure 1. Function Structure Diagram

Procurement management can query the information of incoming goods, and maintain the good information. Refund management and stock management are two departments, but considering the management situation of small and medium-sized supermarkets, system put it in stock management to operate together.

Distribution management is the process used to oversee the movement of goods from supplier to manufacturer to wholesaler or retailer and finally to the end consumer. Numerous activities and processes are involved, including raw good vendor management, packaging, warehousing, inventory, supply chain, logistics and sometimes even block-chain.

The key to an efficient and profitable business is total visibility into the stock process from start to finish, along with management tools to help you maintain optimal stock levels year-round. The good stock management system can help increase profitability and reduce waste. Stock is usually one of the largest assets a company owns, which is why stock mismanagement is one of the top reasons small businesses fail.

The overall workflow in inventory management system consists of two main parts. Headquarter can manage buying from the suppliers and the main stock, order to the suppliers, distribute to the branches if there is need to distribute item. In the parts of buying items from the suppliers and distributing items to the branches, the seasonal multiple regression and the least squares regression method are used to predict how much items are bought from

the suppliers and how much items are distributed to the branches on the stock-out data of each branch. The overall workflow is shown in Fig (2).

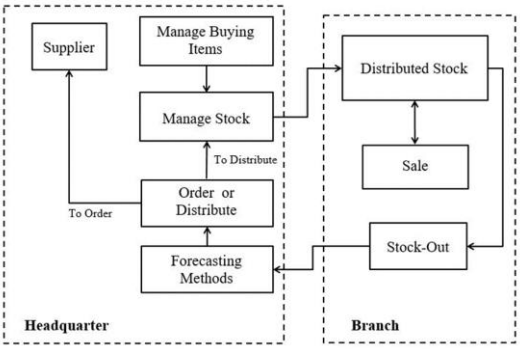


Figure 2. Overall Workflow of Proposed System

A supermarket's purchase list which consists of many branches is updated with things that have been purchased from suppliers. The procedure for the distribution of these refills to branches is pending their storage in stock. Distribution of inventory is made to branches in need of refilling. The database at headquarters is filled with recorded sales and distribution information.

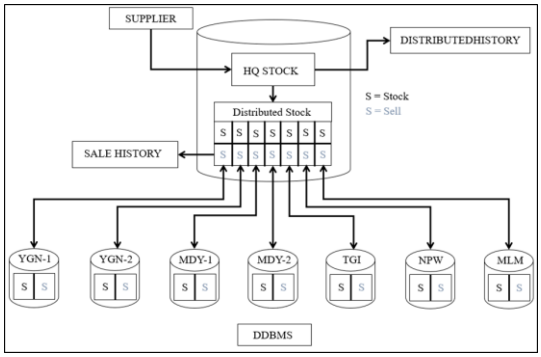


Figure 3. Workflow of Proposed Distributed Database Management Model

Additionally, stores keep a daily sales record for each item as well as dispersed inventory information in their own DB. Headquarter does not distribute inventory to the DBs of the branches, only to its own DB. To get the inventory distribution from the headquarter office, branches must connect to the DB of Headquarter. The sales records of the products supplied to clients each day must be sent by the retailers to Headquarter. The workflow of this process as proposed model is shown in Fig (3).

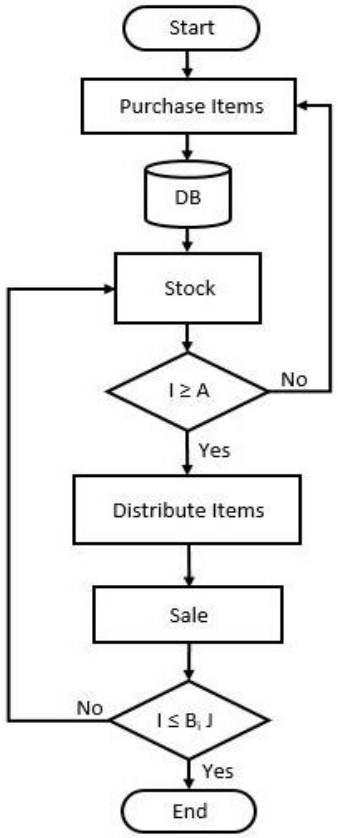


Figure 4. Flow Chart of Proposed Model for Inventory Management System

In Fig 4, the flow chart diagram of an inventory management system is shown. Items are purchased from suppliers and purchased inventory is entered into the Oracle database. Then the stock in the DB is distributed to the branches according to a small number of items. If the quantity of items (I) is more than or equal to distributed amount (A), items are distributed to the branches. If not, more items have to be purchased from suppliers. When the number of items (I) is sufficient, the branches sell the items. If the number of items (I) is less or equal to minimum amount (J) of the branches (B_i), there is need to be replenished. Headquarter has to make sure to add any more goods to stock if there aren't enough available at the time of sale.

4. EXPERIMENTAL RESULTS

4.1. Acquisition

The system starts by analysing the supermarket's inventory needs, such as what products they sell, how many products they sell, how they sell products, how frequently they reorder products, and what their storage capabilities are. Using these needs, the tables are created in Oracle DB to store the collected data. These tables can be categorized by the type of action, such as inventory, sales, and ordering. In any organization, institution or any system of operation there is always an input into the system, which keeps a system going, if the input is wrong definitely the output will be wrong. This design is meant to handle data about a particular product or stock in the supermarket. The following tables mainly formed in Oracle DB are belonged to Headquarter DB.

Table 1. The Main Stock Table

Column Name	Data Type
ID	NUMBER
BARCODE	NVARCHAR2(20 CHAR)
ITEMNAME	NVARCHAR2(100 CHAR)
CATE	NVARCHAR2(30 CHAR)
QTY	NUMBER
PPRICE	NUMBER
SALE_PRICE	NUMBER
DISCOUNT	NUMBER
MIN_QTY	NUMBER
ACTION_DATE	DATE
REMARK	NVARCHAR2(300 CHAR)

Table 2. Purchase Items Table

Column Name	Data Type
ID	NUMBER
BUYDATE	DATE
B_VNO	NVARCHAR2(50 CHAR)
BARCODE	NVARCHAR2(20 CHAR)
ITEMNAME	NVARCHAR2(100 CHAR)
CATE	NVARCHAR2(30 CHAR)
BUY_QTY	NUMBER
PRICE	NUMBER
TPRICE	NUMBER
SUPPLIER	NVARCHAR2(50 CHAR)
ACTION_DATE	DATE
REMARK	NVARCHAR2(300 CHAR)

Table 3. Distributed Stock Table

Column Name	Data Type
DID	NUMBER
DIS_DATE	DATE
BARCODE	NVARCHAR2(20 CHAR)
ITEMNAME	NVARCHAR2(100 CHAR)
CATE	NVARCHAR2(30 CHAR)
DIS_QTY	NUMBER
SALE_PRICE	NUMBER
DISCOUNT	NUMBER
MIN_QTY	NUMBER
BRANCH	NVARCHAR2(20 CHAR)
ACTION_DATE	DATE
REMARK	NVARCHAR2(300 CHAR)

4.2. System Implementation

In this section show the system implementations by using ASP.NET + Microsoft.NET Framework technology and Oracle DB. An ASP.NET application can be developed to connect to the Oracle DB and data are retrieved from the tables. Codes are developed in ASP.NET to distribute data from main server to branches of the supermarket using the Oracle DB, ensuring that the correct inventory levels are maintained at all locations. The

branches may belong to different geographic locations. In spite of the distance between each other, the distributed data can also locate in Head Quarter (HQ) and in each related branch. Then the HQ assigns the latest appropriate warehouse to supply goods based on the branch's location and delivery date to reduce transportation costs and so on. The system develops code in ASP.NET to perform CRUD operations (create, read, update, delete) on the Oracle DB, allowing the supermarket to manage inventory levels and sales data efficiently. The system also develops code in ASP.NET to receive data from the supermarket's HQ, such as updated inventory levels or new products, and update the Oracle DB accordingly.

4.3. Implementation Results

In this section show the results of the system implementation of distributed inventory management system for supermarket with the screen records. The users (managers) firstly log in the system through the log in page, shown in Fig (5). All items entering the stock have been registered. If the item is new or has not yet been registered, it must be registered on the registration form, as shown in Fig (6). The inventory of items purchased from suppliers will then be entered into the purchasing system as shown in Fig (7). Once this is done, those items will be registered to the main stock as well. How it works is shown in Fig (8).

The inventory on the main server will be distributed to the branches according to their inventory requirements. There are two kinds of distribution: individual distribution and multiple distribution of inventory items. In the individual distribution, it is intended to distribute the item list because the item type was entered incorrectly and the item list remains to be distributed. Multiple distribution from the main server to branches involves initial multiple distribution of the inventory and subsequent multiple distributions as needed.

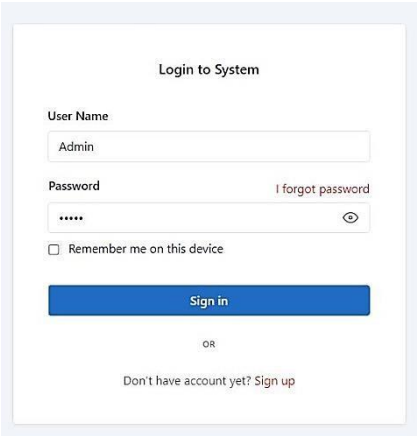


Figure 5. Log In Form

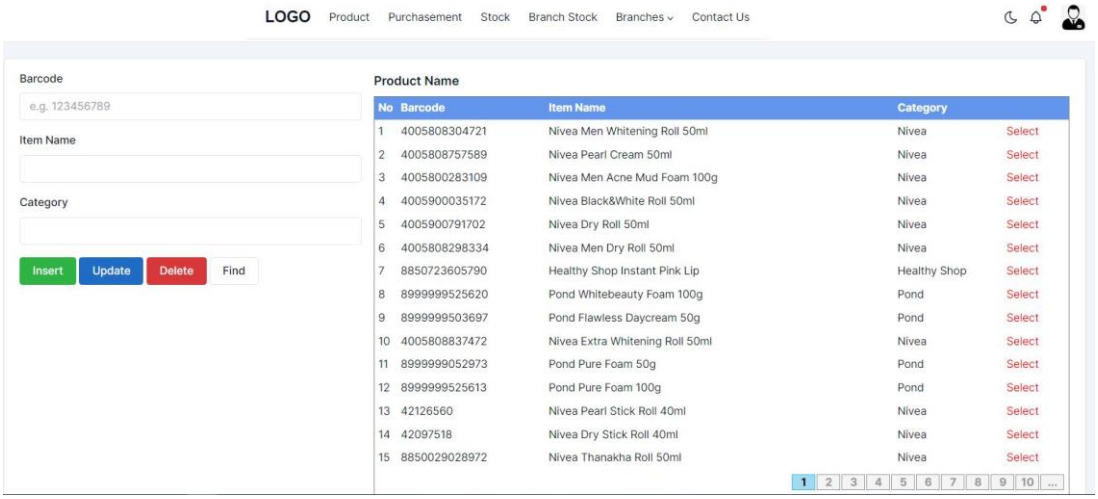


Figure 6. Registration of Product Page in Application

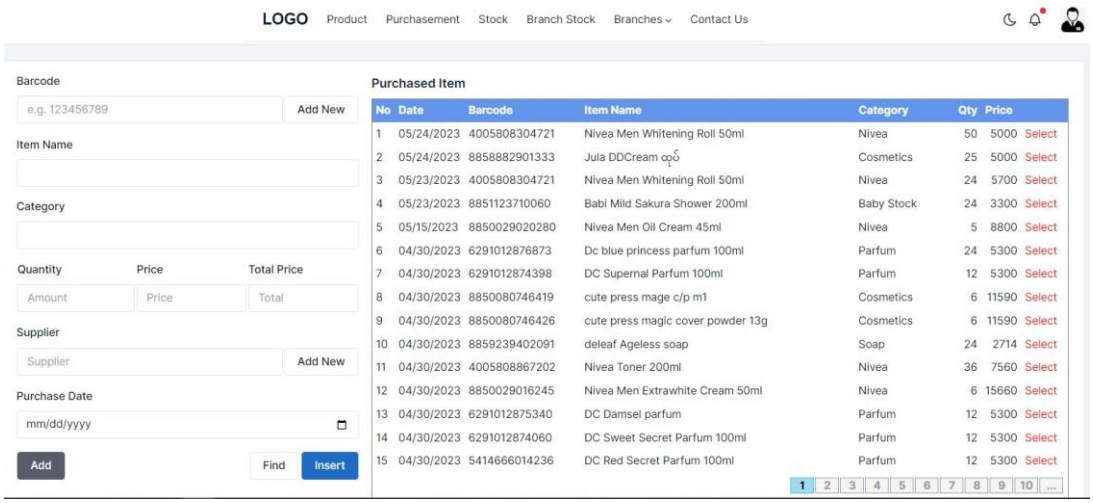


Figure 7. Purchasing Product Page in Application

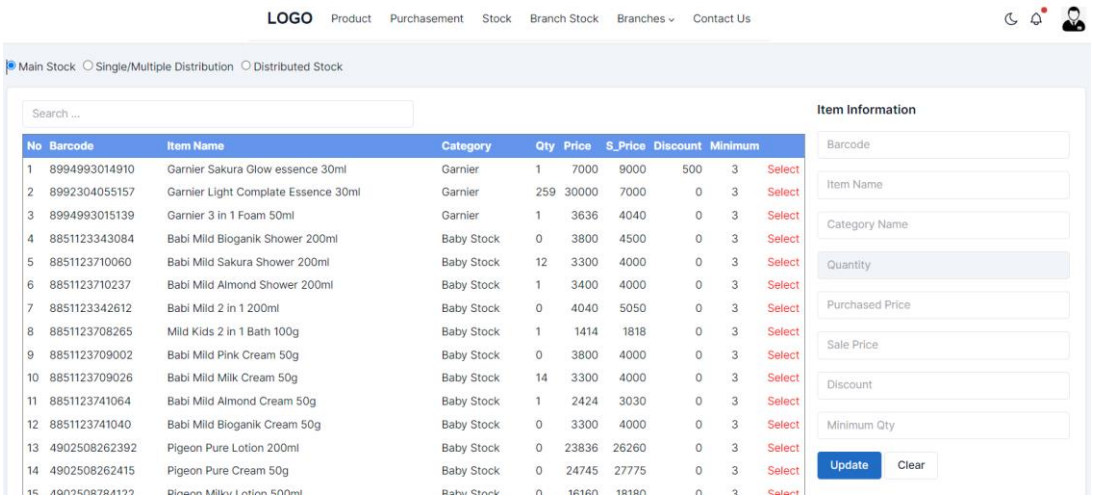


Figure 8. Distributing Product Page in Application

The system includes the control and management of inventory distribution from Headquarter to the branches, and the activities of selling the items from the branches and reporting back to Headquarter.

3. CONCLUSIONS

This researched paper has showcased the integration of cutting-edge technologies to address the intricate challenges of inventory management in a supermarket setting. It showcases an integrated approach, combining DDBMS methods, ASP.NET, and Oracle Database, can address the complexities of data management, real-time information dissemination, and user interaction. Through the examination of this paper, it is evident that utilizing Distributed DBMS methods in inventory management systems offers several benefits. These include scalability to handle large volumes of data, fault tolerance to ensure system reliability, real-time data access for accurate inventory tracking, and improved performance through distributed data storage and processing. Additionally, distributed DBMS methods enable efficient data synchronization across multiple locations, concurrent access control for seamless user interactions, and enhanced data integrity for reliable decision-making.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisors, Dr. Zaw Ye Aung, for their guidance, support, and encouragement throughout the entire process. Their

mentorship and expertise were invaluable in helping me to shape the direction of my research and to bring my ideas to fruition.

REFERENCES

[1] Arsan T., Bas, E_kan , E. Ar _ Z. and Bozkus (2013), "A Software Architecture for Inventory Management System", DOI: 10.1007/978-1-4614-3535-8_2.

[2] Nazar Sohail, Tariq Hussain Sheikh (2018), "A Study of Inventory Management Case Study," *Journal of Advanced Research in Dynamical & Control Systems*, Vol. 10, 10-Special Issue, pp. 1176-1190.

[3] J. Y. Chen, C. Y. Lin, H. C. Zeng (2014), "Optimization of Oracle Database Application System," *Journal of Information Security and Technology*, vol. 4, no. 12, pp. 73-74.

[4] Abisoye Opeyemi A., Boboye Fatoba (2013), "Design of a Computerized Inventory Management System for Supermarkets", *IJSR*, India Online ISSN: 2319-7064, Volume 2 Issue 9.

[5] Monica Kadam, Shubhangi Tambe, Pooja Jidge and Ekta Tayade (2014), "Cloud Service Based On Database Management System", *Journal of Engineering Research and Applications*, Vol. 4, Issue 1(Version 3), January, pp.303-306.

[6] Liling Xia (2011), "The Design and Implementation of Distributed Inventory Management System Based on the Intranet Architecture", *International Conference on Information and Automation*, Shenzhen, China June.

[7] Yongchang Ren, "Design and Implementation of Supermarket Management System", *College of Information Science and Technology*, Bohai University, Jinzhou, 121013, China.

Authors

Biography

Nay Bala, a graduate of B.C.Sc., was commissioned from University of Computer Studies, Yangon (UCSY)) in 2010. He earned his M.I.C.Sc at Moscow Institute of Electronic Technology (M.I.E.T) in 2015. Now, he is a Ph.D. candidate of Computer Science Department, Higher Education Center.

Dr. Zaw Ye Aung, a graduate of B.C.Sc., was commissioned from University of Computer Studies, Yangon (UCSY)) in 2002. He earned his M.E (IT) from Bauman Moscow State Technical University, his PhD in 2012 and D.Sc. (Hons) 2018. Now, he is serving as a lecturer in the Department of Computer Science in Higher Education Center.

Dr Htin Kyaw, a graduate of B.C.Sc, was commissioned from the Defence Services Academy in 2005. He earned his M.I.C.Sc and PhD at the Moscow Institute of Electronic Technology (M.I.E.T) in 2015. Now, he is serving as an assistant lecturer at the Department of Computer Science in the Defence Services Academy (DSA).

DEVELOPMENT AND IMPLEMENTATION OF SBIR FOR SKETCH FACES USING DLIB

Hayma Thida Kyaw[2], Myo Min Hein[2], Ye Wint Mg Mg[3]

[1,2,3]Department of Computer Science, Higher Education Centre (Pyin Oo Lwin)

haymathidakyaw@gmail.com [1], myominhein48@gmail.com [2] , linwai46@gmail.com [3]

ABSTRACT

The development and implementation of SBIR for sketch faces using Dlib based on facial recognition is attempted in this paper. The frontal sketch face images used in this study are free of any impediments. About 200 sketch faces were used in this study as the primary source of data for the machine learning technique. An application can define the front sketch face shape, feature extraction performance, and "Similarity" (matching face data) during the identification step in order to search face images. This system can be used to reduce the number of criminal cases that arise today. The experimental findings show how effective our system is at identifying and detecting faces in criminal cases.

KEYWORDS

Dlib, Face Recognition, Face Detection, Feature Extraction, Forensci Sketch , SBIR,

1. INTRODUCTION

Using the extraction features in machine learning approaches, this research suggests a face recognition system. It suggests including the image's extracted features, including color, texture, form, and shape coordinates. There are numerous face recognition techniques available today, including Principal Component Analysis (PCA), Gabor Wavelet Transform, Sketch Based Image Retrieval (SBIR), and others.

This study[1] examined the idea of SIFT matching, used Euclid distance to compare significant points, and employed RANSAC to weed out mismatches. It locates extreme points in scale-space and determines their coordinate, scale, and orientation, which ultimately give rise to a description. This study compares various outcomes produced using various ratio thresholds and ultimately settles on 0.6 as the ideal value taking into account the proper balance of matched points and matching precision.

The system achieves a 95.42% accuracy in face recognition utilizing the Gabor Wavelet - GW, Wavelet Transformation - WT, and Principal Component Analysis - PCA methods [2]. In order to recognize the pictures in the lower dimension, which yields results of nearly 82% in [2], an extended PCA method has been suggested. Wavelet Transform methods and Clustering Support Vector Machine work together to identify features with an accuracy of 90% in [3].

In comparison to sketches, which are typically black and white or grayscale, photos usually contain more color information. Therefore, feature matching with color could possibly produce improved results if color is a key feature for matching photos. In this manuscript, features like edges, texture, and shape may be more significant than color when it comes to sketch matching. Finding matching drawings in this situation might be easier with feature matching without color.

2. AIM

The objective of manuscript is to create an algorithm and implement SBIR for sketch faces using Dlib, enabling the comparison of facial image similarity searches.

3. RELATED WORK

The manuscript[2] uses a combination of SIFT and SURF feature descriptor to match composite sketches to mugshot photographs. The paper presents a novel feature-based method that compares sketches and mugshot images to determine how similar they are. The features extracted using SURF and SIFT are then matched using the nearest neighbor algorithm. There are many difficulties in this forensic sketch-focused work. This area of study must be developed further in order to create effective systems that bridge the gap between two areas entirely. With an accuracy of 95.45% at rank 50, this combined approach using SIFT and SURF-based features did well.

The creation and implementation of SBIR for faces utilizing the NVIDIA Jetson TX2 is the goal of this work[3], which is based on face recognition. Frontal face photos without any obstructions were used as the research's data. This system can be used to reduce the number of criminal cases that arise today. The facial identification and recognition capabilities of this algorithm are quite good for opening doors. Additionally advised is testing the image under various lighting conditions to gauge system performance.

4. THEORITICAL BACKGROUND

4.1 Sketch Based Image Retrieval

Instead of utilizing standard keyword-based queries or image descriptions, sketch-based image retrieval (SBIR) technology enables users to search for photos using hand-drawn sketches or preliminary drawings as input. To identify images that are similar, SBIR systems examine an image's visual characteristics, such as color, texture, shape, and spatial arrangement, and compare them to characteristics of other images in a database.

Features are extracted using SBIR. The features are then indexed using the appropriate and effective structure to respond to user queries. In SBIR, there are various processes. The straightforward one is suggested, as shown in Figure 1.

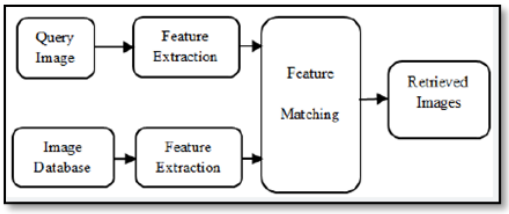


Figure 1. Workflow of SBIR

4.2 Machine Learning

The majority of machine learning-based facial network algorithms are built on deep learning.The field of machine learning is very broad. The study of algorithms and statistical models used by computer systems to carry out specific tasks devoid of explicit directions and in favor of patterns and inference is known as machine learning. It is thought to be a part of artificial intelligence. Machine learning is primarily founded on learning from already existing data and artifacts, such as sales records, search trends, usage patterns, images and images, sounds, text, and so on. Machine learning algorithms can be generally divided into three categories: reinforced learning, unsupervised learning, and supervised learning. In the instance of supervised learning, the system learns based on labels applied to previously classified data. Humans have already classified a collection of data labels. Systems gain knowledge from this input and build a model. There are a plethora of algorithms accessible, and they should be carefully chosen depending on the particular situation.

5. METHODS

5.1 Research Steps

In the system, the input sketch is given as input to the system. The features from the incoming sketch image is extracted by using Dlib method.And then, the features extracted by Dlib are saved in the feature

dataset. After saving features in dataset, the sketch drawn by artist is given as input and the features from sketch are extracted. And then, the features from two input sketch images are matched and finally predict the similarity scores.

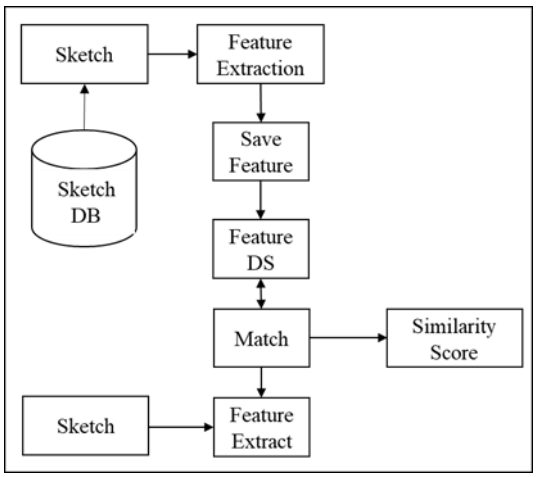


Figure 2. Flowchart of System

5.2 Face Recognition Algorithms

A biometric software tool called facial recognition compares and analyzes patterns based on a person's face characteristics to uniquely identify or authenticate that person. The main applications of facial recognition are in security, however there is growing interest in other applications. Facial recognition technology has actually drawn a lot of attention because it has the potential for a wide range of applications in both law enforcement and other businesses. Various methods of facial recognition are in use, including the adaptive regional blend matching method and generalized matching face detection approach.

The various nodal points on a human face are the basis for how most facial recognition systems work. The algorithm of the approaches used in this study explores the camera features that are available, where each feature is a vector that contains the energy to create a histogram throughout the process of matching during the search process and the introduction of facial photos from the database.

5.3 Evaluation of Face Image Searching

The cosine distance formula, which can be stated as follows, is used to calculate the degree of similarity between a database-stored facial image and the query face image (the reference face).

$$d(p,q) = \cos(\theta) = (q.p)/(\|p\|\|2\|\|q\|\|2\|$$

The Waterfall model is the application development approach employed in this study. It comprises testing, maintenance, and implementation. While the system development life cycle is used to explain the system model for processes. It consists of the general process model's requirements, analysis, design, and flowchart.

5.4 Dlib Facial Landmark Detector

Checking the condition of the face is the first thing that needs to be done. The literature has studied a variety of approaches for assessing the condition of the face. One of these techniques locates the face region and searches for a black circle to see if the pupil is visible. Another technique locates the face region and searches for white pixels. If there are more white pixels than a certain number, we can say that a face is present.

In a short while, artificial intelligence will rule the globe. Face applications allow for the recognition of faces in digital or video pictures. It is an effective illustration of computer vision. The faces must be correctly identified in order to perform identification with greater accuracy and confidentiality. Applications employ face detection technology to identify people in digital photos and movies. Additionally, detecting merely the face won't be helpful. We need more details about the face, such as whether someone grins, laughs, or exhibits dimples when smiling, etc. In conclusion, face expressions convey a lot of information. With the use of facial

landmarks, we are able to learn more about the face.

When locating and displaying prominent areas or facial features on a person's face, facial landmarks are employed, such as:

- Eyebrows
- Eyes
- Jaws
- Nose
- Mouth etc.

Our eyes, lips, nose, and other distinguishing features on our face can all be used to identify us. We actually receive a map of the points that surround each feature when we utilize Dlib's methods to find these features. The following features can be located on a map made up of 67 landmark points:

Figure 3. Face landmark detection

- Jaw Points = 0–16
- Right Brow Points = 17–21
- Left Brow Points = 22–26
- Nose Points = 27–35
- Right Eye Points = 36–41
- Left Eye Points = 42–47
- Mouth Points = 48–60
- Lips Points = 61–67

Using Dlib's facial landmark detector, we will monitor the subject's eyelid movements. This will be used to determine the condition of the eye for each frame as well as other determinations like the number of blinks and the subject's level of drowsiness. To get to our application, we will take the following procedures. After describing the procedure, we will examine how it is carried out in the code along with thorough justifications and preliminary outcomes.

5.5 Experimental Result

The experimental result and comparison results are as shown in the following. The similarity scores is calculate by using MAE, PSNR and SSIM for traditional image processing and modified GAN.

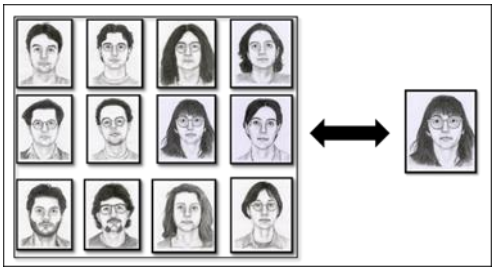


Figure 4. Result of System

Table1. Comparison Result of Similarity Score

Method	PSNR	MAE	SSIM
Ground Truth	inf	1	1
Traditional	30.2832	60.54	0.654
Our GAN	35.6402	74.67	0.750

The features from the input sketch image are extracted by using Dlib function and the result is as shown in the figure.

Figure 5. Results Using Dlib

After extraction the features from the input image by using Dlib, the features are illustrated in the following table.

Table 2. Features Result with Points

Jaw Points	x0-x16 =243-491	y0-y16=281-285
Right Brow Points	x17-x21=266-350	y17-y21=261-258
Left Brow Points	x22-x26=383-472	y22-y26=258-265
Nose Points	x27-x35=364-394	y27-y35=280-365
Right Eye Points	x36-x41=288-302	y36-y41=280-284
Left Eye Points	x42-x47=401-417	y42-y47=288-289
Mouth Points	x48-x60=302-308	y48-y60=398-401
Lips Points	x61-x67=346-344	y61-y67=404-419

6 CONCLUSION

The research suggested a brand-new facial recognition technology. By feature extraction, it may be determined from the findings of the research and testing that have been done. Additionally, this model can identify faces at a variety of resolutions. It may be concluded that employing categorization while processing huge image data yields more accurate results than doing it manually.It is also suggested to test the image capture in different lighting.

ACKNOWLEDGMENT

We are thankful to express my sincere thanks to Dr. Soe Win Myint, Head of Department of Computer Science, and my honorable teachers in Higher Education Centre, for giving this opportunity to read this paper and helping me throughout this research paper.

REFERENCES

[1] Lisa Fan, Jason Krone, Sam Woolf, Sketch to Image Translation using GANs, Project Report at Stanford University, 2016.

[2] Nischal Tonthanahal1, Sourab B R, Dr Sharon Christa, Forensic Sketch-to-Face Image Transformation Using CycleGAN, Dept. of Information Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru.

[3] Shikang Yu, Hu Han, Shiguang Shan, Antitza Dantcheva, Xilin Chen, Improving Face Sketch Recognition via Adversarial Sketch-Photo Transformation, FG 2019 - 14th IEEE International Conference on Automatic Face and Gesture Recognition, Lille, France ,May 2019.

[4] Vinay,Aditya Lokesh, Vinayaka R Kamath,K N B Murtht and S Natarajan, Sketch to image using CNN and GAN, CPRIM PES University Bengalura, India, 2019 IEEE.

Authors

Biography

Hayma Thida Kyaw received Master of Computer Science (M.C.Sc.) degree from Higher Education Center (HEC), in 2019. Now she is a Ph.D. candidate of Computer Science Department, Higher Education Center (HEC).

Dr. Myo Min Hein earned his Master's Degree from Russian State Technological University (MEPHI), (Russian Federation) in 2010. He obtained his Ph.D. (Computer Science) from HEC in 2018. Now he is an assistant lecturer of Department of Computer Science in HEC.

Dr. Ye Wint Mg Mg earned his Master's Degree from Moscow Engineering Physics Institute (MEPHI), Moscow in 2009. He obtained his Ph.D. (Computer Science) from HEC in 2015. Now he is an assistant lecturer of Department of Computer Science in HEC.

PERFORMANCE COMPARISON OF MEMORY ALLOCATION ALGORITHMS

Min Min Su[1], and Wuithmone Oo[2]

[1] Faculty of Computer Science, University of Computer Studies, (Mandalay)

[2] Faculty of Computer Science, Myanmar Institute of Information Technology

minminsu1976@gmail.com, wuithmoneoo1989@gmail.com

ABSTRACT

Managing memory is an essential feature of any computer device. And contiguous memory allocation is a memory management technique used by operating systems to allocate memory to processes in contiguous blocks. In this technique, a process is allocated a single block of memory that is contiguous or adjacent to each other. This ensures that memory is efficiently utilized, with minimal fragmentation and wasted memory. Contiguous memory allocation is a widely used technique in modern operating systems and has several advantages, including efficient memory utilization, fast access to memory, and simple management. Allocation of distributed memory has played a very significant role in memory management. This method offers a realistic simulation and explores the comparison of three major allocation algorithms for memory in multiple partition contiguous allocation schemes using a measured model on three separate sets of process sizes. Different allocation algorithms for memory were designed to effectively organize the memory. Allocation algorithms are implemented to decide the space that can be allocated a process amongst the available ones in the partitioned memory chunk. The findings of the analyzes indicate that best-fit allowed greater usage of the usable memory space than first-fit and worst-fit.

KEYWORDS

Internal fragmentation, External fragmentation, First-Fit, Best-Fit and Worse-Fit.

1. INTRODUCTION

Allocation of dynamic memory is when an executing program requests that the operating system give it a main memory block. We will be including four sets of processes and bulks on a memory partition in this object. Performance of the algorithms depends on their full memory usage [7].

First-fit, best-fit, and worst-fit algorithms are calculated when allocating processes to a memory under the dynamic allocation strategy. We will use the placement algorithm to efficiently allocate the processes in the main memory. The programmer expressly demands that the machine assign the memory and return the starting address of the dynamic allocation memory. For all data, memory must be allocated Compile-time (static) or Run-time (dynamic) [9].

We will present the demonstration of three algorithms and compare their performance on generated simulated trace [4]. First fit select the first block which can satisfy the request. Best fit search on each allocation for the entire list and choose the smallest block that can satisfy the request and then stop searching if the exact match is found. Choose the largest block for worst fit.

2. RELATED WORKS

Memory management is the process of controlling and coordinating computer memory, assigning portions called blocks to various running programs to optimize overall system performance. Then placement algorithms are implemented to determine the slot that can be allocated process amongst the available ones in the partitioned memory. Memory slots allocated to processes might be too big when using the existing placement algorithms hence losing a lot of space due to internal fragmentation. In dynamic partitioning, external fragmentation occurs

when there is a sufficient amount of space in the memory to satisfy the memory request of a

process. Therefore L. W. Htun et.al describes how to resolve external fragmentation using three allocation algorithms and compare their performance. As a result, both First-fit and Bestfit are better than Worst-fit in terms of decreasing both time and storage utilization [3].

Ledisi G. Kabari, Tamunoomie S. Gogo gives a practical demonstration and analyze the efficiency of three major memory allocation algorithms in the multiple partition contiguous memory allocation scheme using a mathematical model on three different sets of process sizes in terms of percentage total utilization, percentage external fragmentation and percentage internal fragmentation. The result from the analysis shows that best-fit made efficient use of the available memory space better than the first-fit and worst-fit [4].

M. A. Awais gives a comparative analysis of some well-known memory management techniques to highlight issues for real-time systems and innovative techniques suitable for these applications will be argued. As a result, it's found that the larger fragmentation, slow response time, larger allocation, and deallocation time with implementation constraints, it makes traditional dynamic memory allocators like the segregated fit, indexed fit, bitmapped fit, and simple buddy system are feasible and inefficient for real-time system [2].

3. MEMORY MANAGEMENT

The main aim of memory management is to make the most efficient use of the available memory. The problem with the design of Memory Management is shown in figure 1. Memory management is a process of system-level management of computer memory with an example of figure 2. Memory management of the operating system relates to the memory allocated to programs [7, 1].

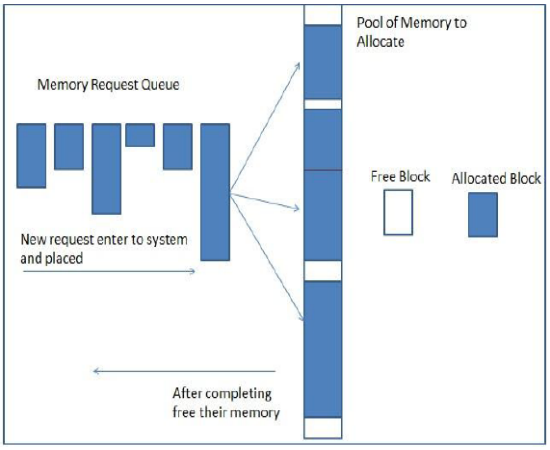


Figure 1. Memory management design problem

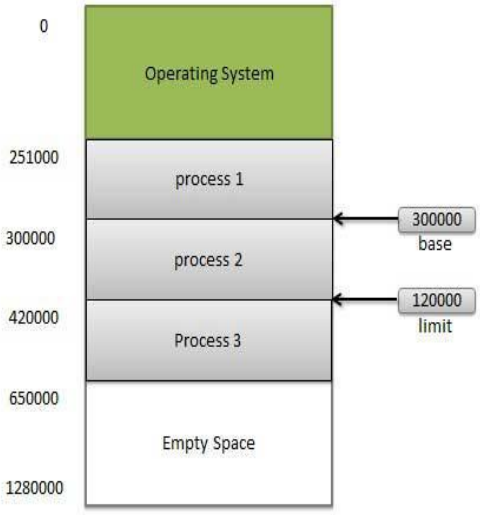


Figure 2. Sample Base Register and a Limit Register in OS

3.1. Dynamic Memory Allocation

Different allocation algorithms for the memory allocation were formulated to effectively form memory. Allocation of dynamic memory is required to manage the available memory. For instance, we may not know the exact memory needed to run the program during compile time. Therefore, resource management choices are taken for the most part at runtime. Contiguous-memory allocation is required to support dynamic memory allocation [7, 8].

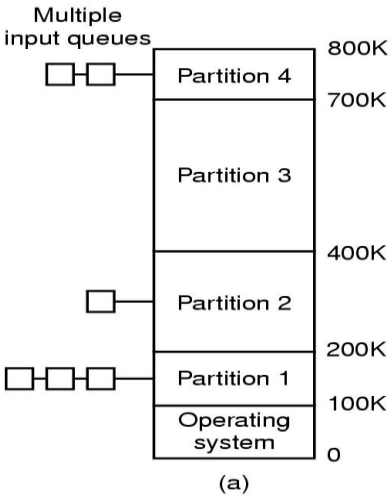


Figure 3. Variable-sized Partitions

3.2. Fragmentation

Fragmentation of the memory may both be internal and external. As systems are loaded and removed from memory, the free space in the memory is broken into small pieces. It occurs when processes can't be assigned to memory blocks often given their limited scale, so storage devices stay unused. It is known as Fragmentation. The continuous memory allocation system suffers from external fragmentation because address spaces are contiguously distributed and when old processes die and new processes are introduced, holes create. The first-time and best-fit memory allocation strategies both suffer from external fragmentation. [11].

3.2.1. Internal Fragmentation

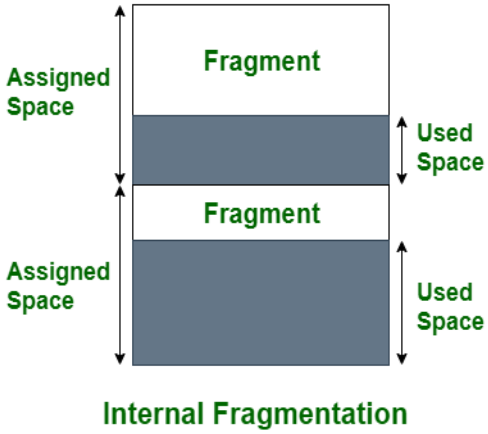


Figure 4. Internal Fragmentation in OS

If the free memory is present within a partition, then it is called "internal fragmentation". Similarly, if the free blocks are present outside the partition, then it is called "external fragmentation". Solution to the "external fragmentation" is compaction. Solution to the "internal fragmentation" is the "placement" algorithm only. Internal fragmentation is the area in a region or a page that is not used by the job occupying that region or page as shown in figure 3. This space is unavailable for use by the system until that job is finished and the page or region is released [6, 11].

3.2.2. External Fragmentation

External fragmentation happens where there is adequate overall memory capacity to fulfil a requirement but there is no contiguous space available. Space is divided into a number of narrow spaces. That problem of fragmentation can be severe. Virtual duplication happens where there is adequate overall memory capacity to fulfil a requirement but the spaces accessible are not contiguous [6, 8]. External fragmentation is unused space between assigned memory regions. Usually, external fragmentation leads in memory regions that are too limited to satisfy a memory request, but if we managed to merge all external fragmentation regions, we would have enough resources to fulfil a capacity request.

4.THREE TYPE OF MEMORY ALLOCATION

Allocators can be classified to attend to the strategy used (first fit, best fit, good fit, etc.) The algorithm's competency is based on its maximum memory utilization. First-fit and best-fit in terms of speed and storage use better than worse [4].

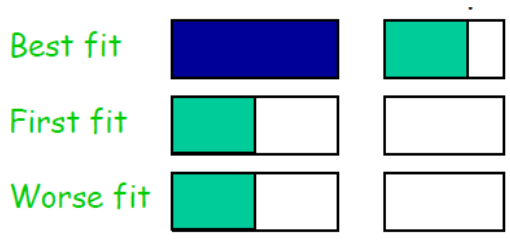


Figure 5. Example of Three Algorithms

4.1.First-fit

For all contrasts the First-fit allocator is used. It will not produce successful time and

fragmentation results but is a source of guidance. First-fit allocator queries the available lists and picks the first block, the size of which is equal to or greater than the size needed. Allocate the first hole, which is wide enough [7, 10].

4.2. Best-fit

Best-fit produces very great results over time on fragmentation but bad results. Best-fit goes deeper in finding the block that better matches the order. Assign the smallest hole that is broad enough; check the whole list when sorted by size and the smallest remaining hole is generated [7, 10].

4.3. Worst-fit memory allocation

In Worst-fit allocation of resources, OS assigns the maximum hole to the process and results in a large volume of resource wastage. Assign the largest hole; look the complete list and generate the largest remaining hole. [5, 10].

5. Implementation Of Memory Allocation Algorithms

A method for overcoming external fragmentation is compaction. The difficulty with compaction is that it is a time-consuming and requires relocation capability. Therefore, different strategies may be taken as to how space is allocated to processes: the common placement algorithms are Firs-fit, Best-fit and Worst-fit.

Operation in First-fit:

a. Input memory chunks with size and processes with size.
b. Set all memory chunks as open.
c. Start by selection each process and check if it can be allocated to current chunk.
d. If process size <= block size if yes then allocate and check for next process
e. If not then keep checking the further chunks.

Operation in Best-fit:

a. Input memory chunks with size and processes with size.
b. Prepare everything memory chunks as open.
c. Start by selection each process and find the minimum chunk size that can be allocated to present process i.e., find min (block-Size [1]), block-Size [2] ,……, block-Size [n] > process-Size [present] it initiate then allocate it to the present process.
d. If not then permission that process and keep trying the additional processes.

Operation in Worst-fit:

a. Input memory chunks with size and processes with size.
b. Prepare entirely memory chunks as open.
c. Start by selection each process and find the maximum chunk size that can be allocated to present process i.e., find max (block-Size[1]), block-Size[2],……., block-Size[n] > process-Size [present] it found then allocate it to the present process.
d. If not then permission that process and keep trying the additional processes.

5.1. Experimental Results of Memory Allocation Algorithms

Experimental Results of Memory Allocation Algorithms are showed in this section. Four sets of processes and sizes on a partition of memory are used for the comparison. The First set of memory is separated into seven pieces as given in below, having a total memory size of 2500 KiloBytes and different process size.

Set 1 processes and their sizes

Block-Size = {100, 600, 200, 300, 500, 200, 600};
Process-Size = {212, 317, 112, 426, 308, 519, 205};
Output results for this chunk and process are shown in table 1 and figure 6.

Table1. First Set Output Results

Process No.	Process Size	Block no.		
		BF	FF	WF
1	212	4	2	2
2	317	5	2	7
3	112	5	3	5
4	426	2	5	Not Allocated
5	308	7	7	2
6	519	Not Allocated	Not Allocated	Not Allocated
7	205	7	4	5

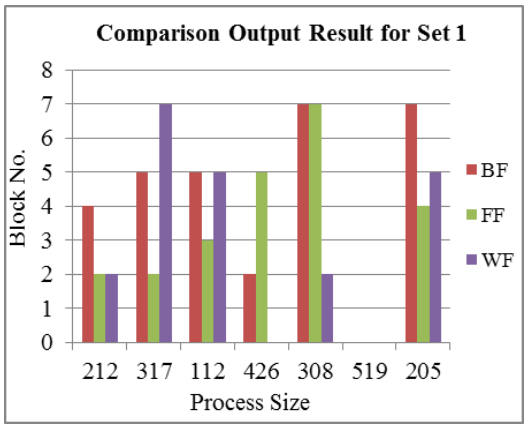


Figure 6. Comparison of Three Allocation Algorithms for Performance of Set 1

Set 2 processes and their sizes

Block-Size = {200, 500, 150, 300, 250, 200, 400};

Process-Size = {212, 117, 112, 326, 308, 219, 175};

Output results for this chunk and process are shown in table 2 and figure 7. The Second set of memory is separated into seven pieces as set in below, taking a total memory size of 2000 KiloBytes and different process sizes.

Table 2. Second Set Output Results

Process No.	Process Size	Block no.		
		BF	FF	WF
1	212	5	2	2
2	117	3	1	7
3	112	1	2	4
4	326	7	7	Not Allocated
5	308	2	Not Allocated	Not Allocated
6	219	4	4	2
7	175	2	2	7

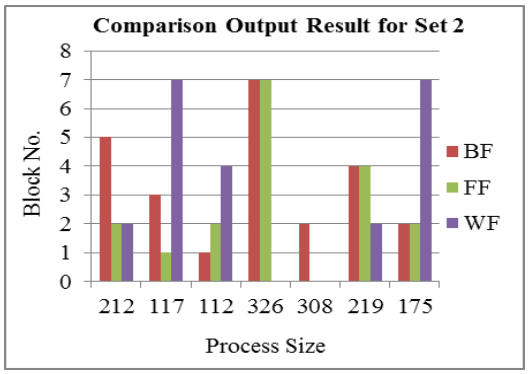


Figure 7. Comparison of Three Allocation Algorithms for Performance of Set 2

Set 3 processes and their sizes

Block-Size = {100, 600, 200, 300, 500, 200, 600};

Process-Size = {212, 317, 112, 320, 308, 330, 205};

Table 3. Third set Output Results

Process No.	Process Size	Block no.		
		BF	FF	WF
1	212	2	2	4
2	117	7	2	5
3	112	5	3	5
4	326	2	5	2
5	308	5	7	7
6	219	Not Allocated	Not Allocated	Not Allocated
7	175	4	4	2

The third set of memory is separated into seven pieces as assumed in below, having a total memory size of 2500 KiloBytes and different process size. Output results for this chunk and process are shown in table 3 and figure 8.

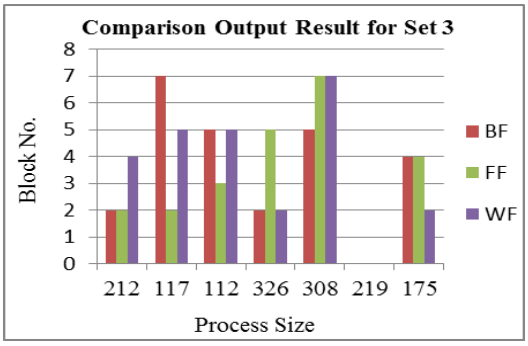


Figure 8. Comparison of Three Allocation Algorithms for Performance of Set 3

Set 4 processes and their sizes

Block-Size = {300, 600, 200, 300, 500};

Process-Size = {212, 217, 112, 220, 308, 250};

Table 4. Fourth set Output Results

Process No.	Process Size	Block no.		
		BF	FF	WF
1	212	1	1	2
2	217	4	2	5
3	112	3	2	2
4	220	5	2	1
5	308	2	5	Not Allocated
6	250	5	4	4

The fourth set of memory is separated into six pieces as specified in below, having a total

memory size of 1900 KiloBytes and different process size. Output results for this chunk and process are shown in table 4 and figure 9.

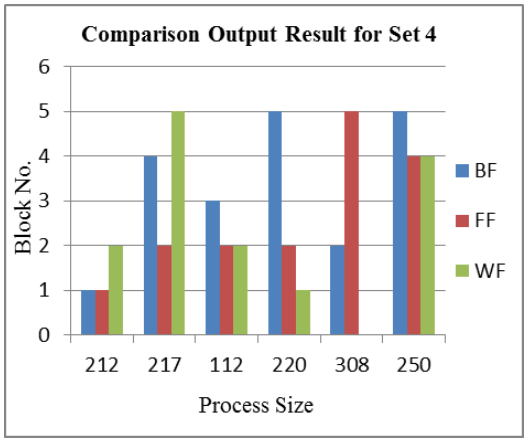


Figure 9. Comparison of Three Allocation Algorithms for Performance of Set 4

According to the implementation, best-fit is efficient use of the available memory space more than that of first-fit and worst-fit. We have been seen that Best-fit algorithm is the best among three placement algorithms.

6. CONCLUSION

Modern operating systems provide efficient memory management Main memory management is very important in operating system. Simulations have shown that both First-fit and Best fit are better than Worst-fit in terms of decreasing both time and storage utilization. Neither First-fit nor Best-fit is clearly better in terms of storage utilization. Because First-fit is the fastest search as it searches only the first block i.e. enough to assign a process. Therefore First-fit can decrease time consuming. Among three algorithms, Best-fit algorithm makes the most efficient use of memory. Because it looks for the smallest block that will satisfy the requirement, it guarantees that the fragment left behind is as small as possible. But best fit can be the worst performer. The next-fit will more frequently lead to an allocation form a free block at the end of memory. The result is that the largest block of free memory which appears at the end of the memory space, is quickly broken up into small fragments. Thus, compaction may be required more frequently with it. Therefore, next-fit tends to produce worse results than the other two algorithms. This system has shown a detail explanation and investigation of the three major memory allocation algorithms in memory allocation schemes. This system can extend comparison of Internal Fragmentation (IF), External Fragmentation (EF), and memory utilization in term of three type of allocation.

REFERENCES

[1] Z. Ali, “Memory Management Strategies for Different Real Time Operating Systems”, https://www.researchgate.net/publication/309771868

[2] M. A. Awais, “Memory Management: Challenges and Techniques for Traditional Memory Allocation Algorithms in Relation with Today's Real Time Needs”, *Internal Journal of Multidisciplinary Sciences and Engineering* ,Vol.7, No.3, March 2016, pp. 13-19.

[3] L.W. Htun, M. M. M. Kay, and A. A. Cho, “Analysis of Allocation Algorithms in Memory Management”, *International Journal of Trend in Scientific Research and Development (IJTSRD)*, Vol.3, Issue 5, August, 2019, PP 1985-1987.

[4] L. G. Kabari, and T. S. Gogo, "Efficiency of Memory Allocation Algorithms Using Mathematical Model", *International Journal of Emerging Engineering Research and Technology*, Vol. 3, Issue 9, September, 2015, PP 55-67.

[5] S. Madnick and J. Donovan. “Operating Systems”, *McGraw-Hill Book Company*. ISBN 0-07-039455-5. Pp: 105 – 208, 1974.

[6] Meenu, V. Dhull, Monika, “Computational Study of Static and Dynamic Memory Allocation”, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 5, Issue 8, August 2015.

[7] N. V. Patil and P. S. Irabashetti, “Dynamic Memory Allocation: Role in Memory Management”, *International Journal of Current Engineering and Technology*, Vol.4, No.2, April 2014.

[8] W. Stallings, “Operating System Internals and Design Principles”, March 20, 2017.

[9] A. Silerschatz, P. B. Galvin, and G. Gange, “Operating System Concepts”, *John Wiley & Sons, INC.*, January 1, 2002.

[10] IBM Corporation (1970). IBM System/360 Operating System: Concept and facilities [online]

[11]https://www.geeksforgeeks.org/difference-
between-internal-and-external- fragmentation/

DEEP LEARNING BASED POINT CLOUD CLASSIFICATION USING A LOW-COST 2D LIDAR IN INVISIBLE CONDITIONS

Aung Zaw Min[1], Aung Aung Hein[2], and Arkar Myint[3]

[1,2]Department of Computer Technology, [3]Department of Computer Science, Higher Education Centre (Pyin Oo Lwin)

aungzawminn1990@gmail.com [1] , moezat198@gmail.com [2] , akmyint317@gmail.com[3]

Abstract

For self-driving cars, security surveillance systems, and many other types of complex automated systems, point cloud classification using LiDAR sensors in invisible conditions is essential. The utilization of three-dimensional models and maps is essential to self-driving cars and a variety of advanced autonomous navigations. Localization and mapping techniques can obtain significantly higher levels of precision by employing dense 3D point clouds which are obtained by expensive three-dimensional LiDAR sensors. But the price of a multi-channel 3D LiDAR with a 360-degree field of view is more than the price of a single-channel 2D LiDAR. Therefore, even though three-dimensional LIDAR have come to be an essential part of autonomous systems, their implementation in smaller robots has been limited by their high cost. In this paper, we have built and classified a three-dimensional point cloud for autonomous systems using a low-cost two-dimensional LIDAR, which rotates via a servo motor. This low-cost architecture, which can generate three-dimensional representations, opens up possibilities for low-cost and small autonomous systems. In addition, we also presented point cloud classification using these constructed 3D point clouds.

Keywords

Arduino, Autonomous Systems, Laser Rangefinder, Laser Scanner, Low-cost, Point Cloud Classification,3D Construction, voxelNet

1. INTRODUCTION

Point cloud classification and detection are increasingly popular in a variety of applications, including face recognition, self-driving vehicles, pedestrian detection, and security surveillance systems. With the development of the Internet of Things [2], more and more image data are collected by various image sensors, video sensors, and point cloud sensors. Users need to know what objects are in the image and where they are located before utilizing image data for more difficult computer vision tasks. Therefore, object detection and classification [3] has always been a hot research direction in the field of computer vision, and its purpose is to locate and classify objects in images or videos, and point clouds.

Over the past decades, 2D and 3D LiDARs have been adopted in a wide range of robots and autonomous vehicles [1,6]. Particularly, night vision sensors are essential for object identification and classification. To capture images or data as an input for object detection can be used various sensors, such as radar, high-resolution cameras, infrared thermal cameras, and LiDAR sensors [5, 4]. LiDAR sensors and infrared thermal cameras are two of them that are frequently used to find and classify objects in the invisible environment. for the fact that it may provide high-quality data input for object detection and classification.

Low-cost sensors are rarely utilized for object detection and classification, and only expensive three-dimensional sensors are typically used. This is because high-end sensors can produce three-dimensional point clouds, but low-end sensors can only produce two-dimensional data. Therefore, using a two-dimensional sensor to identify and categorize objects could be an acceptable approach in low-light situations. However, users must convert two-

dimensional data into three dimensions in order to classify objects using a low-cost two-dimensional LiDAR sensor.

Therefore, in this paper, a method is proposed for transforming three-dimensional point clouds via two-dimensional data for object detection and classification using a low-cost two-dimensional LiDAR sensor. After that, the three-dimensional point clouds are trained for object detection and classification using a Convolutional Neural Network and proposed network. And then, the data will be divided into five different object classes, including humans, trees, unknowns, walls, and corners by using a convolutional neural network and the proposed network.

2. AIM

The aim of this research is to detect and classify objects that cannot be scanned by cameras and that are invisible to humans due to the lack of light.

3. PROPOSED SYSTEM

The primary two phases of our proposed system are presented in this article. They are point cloud training and point cloud classification. In the point cloud training and classification processes, we employ a low-cost 2D LiDAR sensor to collect raw point cloud data in invisible conditions as shown in figure 1. And then, we use polynomial curve fitting to filter out noise or unnecessary data from the raw point cloud data. And, utilizing 3D reconstruction techniques such as finding X, Y, and Z in the Cartesian coordinate system, we reconstructed three-dimensional data from the filtered point cloud using our proposed method. Finally, we used our proposed method to detect and classify the objects into classes such as a wall, human, tree, corner, and unknown.

In the flow chart of proposed system, firstly, the system obtains the input point cloud by using a low-cost two-dimensional LiDAR sensor, LiDAR scanner. The system will next perform the point cloud transformation which is captured by a two-dimensional LiDAR sensor in invisible conditions using a Cartesian coordinate system and a spherical coordinate system.

Figure 1. A low-cost 2D LiDAR

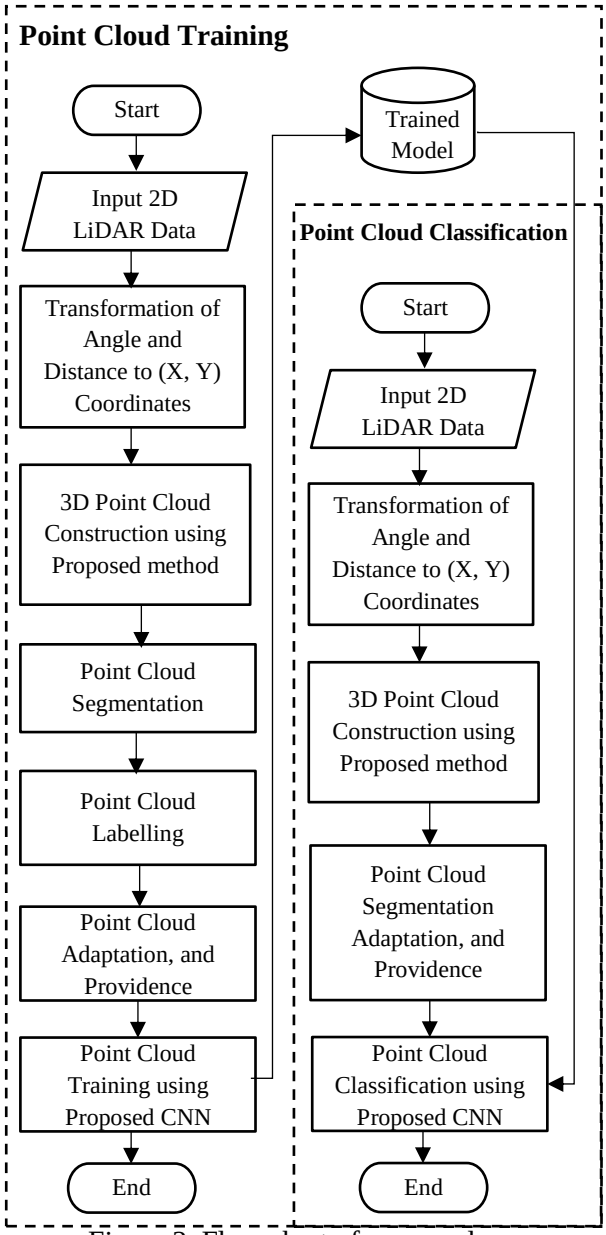


Figure 2. Flow chart of proposed classification system

After transforming the point clouds, the proposed method was used to construct the

point cloud into a three-dimensional point cloud. And then, the three-dimensional point clouds were segmented to label the point cloud and were provided and adapted to enter into the training model. The point cloud has been trained progressively, the proposed convolutional neural network was utilized for point cloud classification in invisible conditions. The flow chart of proposed system is illustrated in figure 2.

3.1 Point Cloud Collection

The LiDAR sensor is a low-cost sensor that can only be received two-dimensional data. Therefore, we need to convert two-dimensional data into a three-dimensional point cloud to perform object detection and classification using a servo motor and Arduino board, as shown in figure 3. LiDAR is installed to collect point cloud data using hybrid equipment for point cloud classification.

Figure 3. 2D LiDAR configuration with the instruments

Figure 4 showed the operations during the acquisition of a -15° to 15° elevation in a room with a 2D LiDAR sensor. The following equation 3 to 8 is three-dimensional point cloud construction which is our proposed method.

$$\alpha' = \begin{cases} \alpha - (d * \Delta x) & \text{if } \theta \\ (d * \Delta x) & \text{if } \theta' \end{cases} \quad (1)$$

$$\Delta x = \frac{\alpha}{90} \quad (2)$$

$$Dist_{pc} = D * \cos(\alpha') \quad (3)$$

$$Z_{pc} = D * \sin(\alpha) \quad (4)$$

$$X_{pc} = Dist_{pc} * \cos(\theta) \quad (5)$$

$$Y_{pc} = Dist_{pc} * \sin(\theta) \quad (6)$$

Where,

α = Pitch of Lidar

θ = Rotational angle

D = Distance from sensor

$Dist_{pc}$ = Actual Distance between sensor and object

Z_{pc} = Height position of point cloud

θ = 0° to 90° / 180° to 270°

θ' = 90° to 180° /270° to 360°

d = 0 to 90 for each quadrant

X_{pc} =The value of X - axis

Y_{pc} =The value of Y – axis

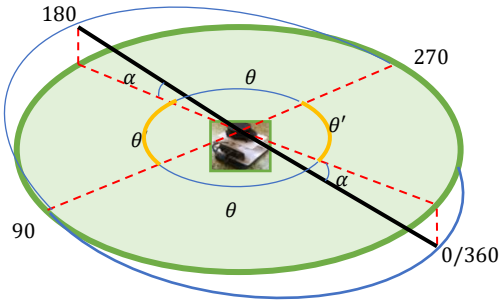


Figure 4. 3D point cloud construction

The LiDAR sensor has a detection range of 0.15 meters to 6 meters and can provide 360-degree ambient depth data. Additionally, it enables both horizontal and vertical field views, assisting in the detection of obstacles as well as objects in the immediate surrounding area. When the LiDAR sensor is set up using the previously mentioned patterns and the objects in the surrounding area are captured, the raw point cloud is shown in figure 5 emerges.

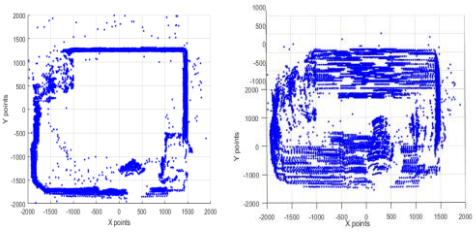


Figure 5. Constructed point cloud using proposed system

3.2 Point Cloud Processing

After the point cloud collection step is complete, presented the pre-processing stage in VoxelNet in this phase. VoxelNet is an end-to-end network that incorporates feature extraction and bounding box prediction. It is a three-dimensional object detection and classification model that reduces the limitations of conventional models by employing revolutionary methods. Our proposed system utilized three different functional structure components. They are point cloud segmentation, point cloud adaptation, and point cloud.

In the point cloud segmentation state, we need to perform point cloud labelling such as the wall, human, tree, unknown, and corner for point cloud training and classification using our proposed method. So, we divided or segmented the three-dimensional point clouds into equal pieces as shown in figure 6. The following equation is the segmentation of three-dimensional point clouds which is our proposed method.

$$M_x = \frac{L(X)}{SB}, M_y = \frac{L(Y)}{SB} \quad (7)$$

$$N_z = \frac{L\,(Z)}{SB} \quad (8)$$

$$SS = M_x * M_y * N_z \quad (9)$$

Where,

SS = Size of segmentation

M_x, M_y = The value of X and Y axis

N_z = The value of height

(X) = Length of X

$L\,(Y)$ = Length of Y

SB = Size of block

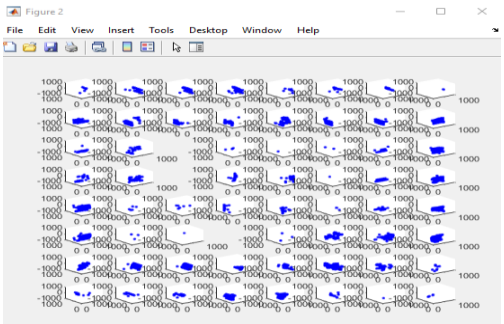


Figure 6. Simulation of point cloud segmentation

After segmenting the three-dimensional point cloud, we will also need to adapt the point cloud for training and classification using our proposed method which is point cloud adaptation. Point cloud adaptation performs to reduce or resizes the dimension of the three-dimensional point cloud for training and classification. The equation of the point cloud adaptation is as shown below:

$$BS = \left\lfloor \left(\frac{SB}{NI}\right) \right\rfloor \quad (10)$$

Where,

BS = block size or point cloud adaptation
SB = size of block
NI = network input size

After performing the point cloud adaptation, If the point cloud has two or more points in the voxel, we use the 3D distance formula to access only one point for training and classification.

After adapting the three-dimensional point cloud, we need to use point cloud providence, which is performed as a binary point cloud, to enter it into the training model for classification. The flow diagram of point cloud providence is illustrated in figure 7.

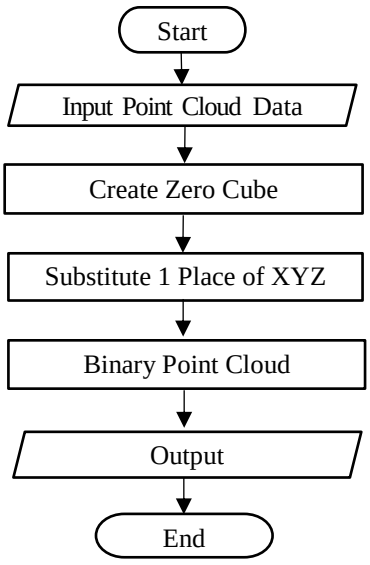


Figure 7. Flow diagram for point cloud providence

In the proposed system of point cloud providence, we localize the three-

dimensional point clouds in X, Y, and Z coordinate using our proposed system. And then, we will create a cube or voxel which is contain zero. We will substitute 1 if the cube or voxel has a three-dimensional point cloud.

In MATLAB R2021b, the point clouds are labelled and resize with a resolution of 500x500x2000 voxel size depending on the point cloud obtained as shown in figure 4.6. The dataset of ground truth labeling is performed by 9635 point clouds of various environments. It contains five object types such as humans, trees, unknowns, walls, and corners. These objects are labelled in the format of .mat file type.

3.3 Proposed VoxelNet Architecture

The input layer of the proposed classification network model obtains a point cloud with a 46x46x46 voxel size. In order to extract the features from the point cloud, we perform a series of two convolution functioning. Figure 8 demonstrates the architecture of the proposed VoxelNet-based object detection and classification method.

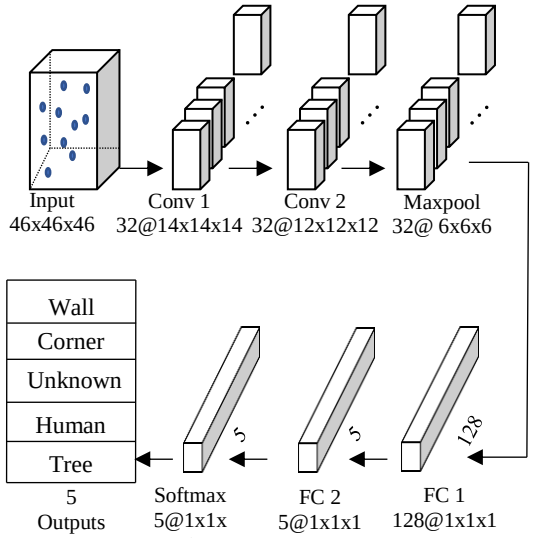


Figure 8. The architecture of VoxelNet

In the proposed classification network model, the input layer receives the point cloud with a size of 46x46x46 voxels. We operate a series of two convolution processes to extract the features of the point cloud.

There are two convolution layers, two fully connected layers, and one max-pooling layers. The input 46x46x46x1voxel passes through the first convolutional layer with 32 feature maps or filters size 5x5x5 padding [0 0 0: 0 0 0] and with a stride of [2 2 2]. And then, we apply LeakyReLu activation function. The resulting point cloud dimensions will be reduced to 14x14x14x32.

Next, the second convolutional layer with 32 feature maps having size 3x3x3, a stride of [1 1 1] and padding [0 0 0: 0 0 0]. The point clouds dimensions' does not change. And, we also use LeakyReLU activation function. This layer has 32 feature maps and the output will be reduced to 12x12x12x32. Then, the max-pooling is applied with a filter size of 2x2x2 and a stride of [2 2 2] and padding [0 0 0: 0 0 0]. The resulting point cloud dimensions will be reduced to 6x6x6x32.

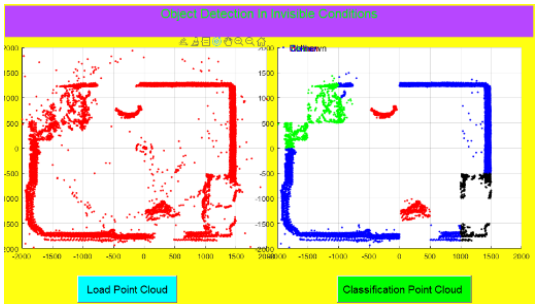
After passing the two Fully-Connected layer and Softmax layer, the final size will be reduced to 1x1x1x5. The classification output layer produces the output corresponding to each group. The output which is related to the 5 classes of object detection and classification such as wall, corner, unknown, human, tree.

4. EXPERIMENTAL RESULTS

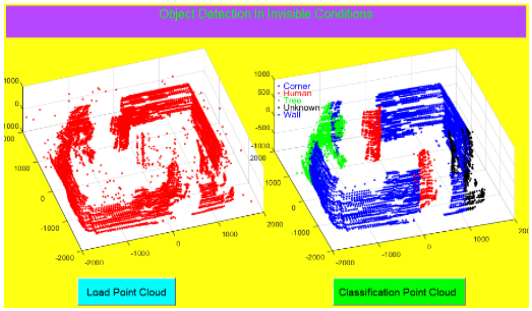
We detect objects using the proposed technique after the object detection and classification model has been trained. Our proposed system performs well in various situations and invisible environments. The experimental results of detection and classification that operate in invisible situations are shown in figure 9. Figure 10 is shown ground truth image for point cloud classification.

In order to make the testing results of our proposed system easier to understand, we use color representation. For example, the blue color represents a wall or corner, the

black color denotes an unknown (The clothes shelf is set to Unknown class.), the red color indicates a person and the green color defines a tree.



(a)



(b)

Fig 9. Result of point cloud classification (a) 2D view, and (b) 3D view

Figure 10. Ground truth image for point cloud classification

We illustrate the classification accuracy of the point cloud in table 1. We tested the point cloud classification's accuracy with 126 scans, including walls, people, trees, unknowns, and corners. The total amount of testing is 3908 frames per type. They are 2207 walls, 198 humans, 647 trees, 352 unknowns, and 504 corners. Our proposed system has an overall accuracy percentage of 97.93%.

Table 1. Accuracy of point cloud classification

Predict classes	Actual Classes: Wall	Human	Tree	Unknown	Corner	
Wall	2151	7	12	5	0	
Human	11	195	1	0	1	
Tree	77	5	644	5	10	
Unknown	59	4	3	352	6	
Corner	0	0	0	0	485	
	2207	198	647	352	504	97.93%

A point cloud classification system is trained to test the proposed point cloud classification system with different voxel sizes and several convolution layers. These different voxel sizes are 32x32x32, 46x46x46, and 64x64x64. These different voxel sizes are trained with 2 convolution layers which involve two kinds of filter sizes and stride sizes in each convolution layer. The four types of environments scanned are denoted by the letters A, B, C, and D. Each scan contains walls, humans, trees, unknowns, and corner classes.

After training the point clouds with different voxel sizes and several convolution layers, we tested the accuracy of the point cloud classification system with 126 scans containing walls, humans, trees, unknowns, and corners in each scan. The total amount of testing is 3908 frames, which include 2207 walls, 198 humans, 647 trees, 352 unknowns, and 504 corner frames. The accuracy of the proposed system is calculated using the confusion matrix.

The accuracy and training times of the point cloud classification for different voxel sizes are illustrated in table 2. These sizes are trained with two convolutional layers. The first layer used filter size 7x7x7 and stride size 2x2x2.

And the second layer used filter size 3x3x3 and stride size 1x1x1.

Table 2. The accuracy of the point cloud classification system

2 Convolution Layers	Enviro-nments	Input sizes	Training Times	Accuracy
Convolition 1 filters 7x7x7, Stride 2x2x2, **LeakyRelu** with scale 0.1 **Convolution 2** filters 3x3x3, Stride 1x1x1, **LeakyRelu** with scale 0.1 **Pooling** filter 2x2x2, Stride 2x2x2 **FC 1** 128 fully connected layer **Relu** Relu **dropout1** 50% dropout **FC2** 5 fully connected layer **softmax** softmax **crossEntropy-Loss** crossentropyex	**A** Total Scan = 37	32x32 x32	313 min	97.21%
		46x46 x46	498 min	99.55%
		64 x 64x64	786 min	94.50%
	B Total Scan = 34	32x32 x32	321 min	95.92%
		46x46 x46	483 min	96.02%
		64x64 x64	791 min	95.54%
	C Total Scan = 38	32x32 x32	309 min	99.52%
		46x46 x46	488 min	99.13%
		64x64 x64	782 min	97.36%
	D Total Scan = 17	32x32 x32	317 min	95.88%
		46x46 x46	485 min	96.32%
		64x64 x64	796 min	95.15%

After testing our proposed system with 126 scans, as shown in table 2, we found that if the voxel size is small, the training time will be less, and if the voxel size is large, the training time will increase. And also found that the 46x46x46 voxel size has better accuracy than other voxel sizes.

In table 3, the accuracy and training of the point cloud classification system for different voxel sizes is shown. The training is used 2 convolutional layers for these sizes. Filter size 9x9x9 and stride size 3x3x3 were utilized in the first convolution layer. The second layer applied stride size 3x3x3 and filter size 2x2x2.

The classification accuracy was tested using 126 scans for four environments, as illustrated by the network structure in Table 3. After testing our proposed system, we found that the training time using filter size 9x9x9 and stride size 3x3x3 was less than the training time using filter size 7x7x7 and stride size 2x2x2. However, the accuracy tested using filter size 9x9x9 and stride size 3x3x3 was found to be less than the accuracy tested using filter size 7x7x7 and stride size 2x2x2.

Table 3. The accuracy of the point cloud classification

2 Convolution Layers	Enviro-nments	Input sizes	Training Times	Accuracy
Convolition 1 filters 9x9x9, Stride 3x3x3, **LeakyRelu** with scale 0.1 **Convolution 2** filters 3x3x3, Stride 2x2x2, **LeakyRelu** with scale 0.1 **Pooling** filter 2x2x2, Stride 2x2x2 **FC 1** 128 fully connected layer **Relu** Relu **dropout1** 50% dropout **FC2** 5 fully connected layer **softmax** softmax **crossEntropy-Loss** crossentropyex	**A** Total Scan = 37	32x32 x32	170 min	84.95%
		46x46 x46	268 min	96.85%
		64x64 x64	418 min	95.95%
	B Total Scan = 34	32x32 x32	156 min	81.21%
		46x46 x46	230 min	94.40%
		64x64 x64	404 min	95.31%
	C Total Scan = 38	32x32 x32	142 min	85.52%
		46x46 x46	226 min	99.32%
		64x64 x64	405 min	97.88%
	D Total Scan = 17	32 x 32x32	142 min	86.62%
		46x46 x46	225 min	94.41%
		64x64 x64	427 min	95.26%

Additionally, it is found that when the proposed system utilizes a small input size, employing a small filter size has greater accuracy than using a large filter size. However, when using a large input size, it will be found that using a large filter size has better accuracy than using a small filter size.

5. CONCLUSIONS

The LiDAR sensor, a low-cost two-dimensional LiDAR sensor, and Voxel Net were used in this research to present state-of-the-art three-dimensional object detection in an invisible environment. Furthermore, in order to convert two-dimensional data into three-dimensional point clouds, we also used an Arduino board and servo motor between the two-dimensional sensor and the central processing unit (CPU). Our proposed system successfully operates three-dimensional form data by directly performing upon sparse three-dimensional point cloud data. In addition, we offer a successful Voxel Net approach that is effective and takes

advantage of point cloud sparseness and parallel processing on a Voxel grid. Additionally, successful results for VoxelNet demonstrate that it performs more difficult tasks, such as the three-dimensional detection and classification of humans, trees, walls, corners, and unknowns.

ACKNOWLEDGMENT

I would like to express my appreciation to Dr. Arkar Myint, Dr. Aung Aung Hein for assisting me with my research. I would also want to express my appreciation to my colleagues for their encouragement and suggestions on the research.

REFERENCES

[1] A. N. Catapang and M. Ramos. “Obstacle detection using a 2d LIDAR system for an autonomous vehicle”. In 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Nov 2017.

[2] Cvar, N, Trilar, J, Kos, A, Volk, M, Stojmenova Duh, E. “The Use of IoT Technology in Smart Cities and Smart Villages: Similarities, Differences, and Future Prospects”. Sensors 2020, 20, 3897.

[3] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D. “Object Detection with Discriminatively Trained Part-Based Models”. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 32, 1627–1645.

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. “nuscenes: A multi-modal dataset for autonomous driving”. In Proceedings of the IEEE CVPR, 2020.

[5] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. “Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather”. In Proceedings of the IEEE CVPR, 2020.

[6] W. Maddern, A. Harrison, and P. Newman. Lost in translation (and rotation): “Rapid extrinsic calibration for 2d and 3d lidars”. In 2012 IEEE International Conference on Robotics and Automation, May 2018.

Authors

Biography

Aung Zaw Min received his M.I.C.Sc at Moscow Institute of Electronic Technology (M.I.E.T) in 2016. Now, he is currently a Ph.D. candidate of the Department of Computer Technology, Higher Education Center (HEC).

Dr. Aung Aung Hein earned his master degree and Ph.D. degree from National Research Nuclear University (MEPhI), (Russian Federation) in 2009 and 2013. Now he is a lecturer of Department of Computer Technology.

Dr. Arkar Myint earned his Master’s Degree from Russian State Technological University (MEPHI), (Russian Federation) in 2004. He obtained his Ph.D. (Computer Science) from HEC in 2009. Now he is an associate professor of Department of Computer Science in HEC.

BRIDGING UML MODELING AND WSDL GENERATION:A MODEL DRIVEN APPROACH WITH FORWARD AND REVERSE ENGINEERING

Wai Phyo Maung[1], Phone Naing[2], and Aung Aung Hein[3]

[1,2] Department of Computer Science,[3]Department of Computer Technology, Higher Education Canter (PyinOoLwin)

waiphyo.maung13@gmail.com[1], kophonenaing7@gmail.com[2], moezat198@gmail.com[3]

ABSTRACT

This paper explores how to combine Model Driven Approaches (MDA) and Unified Modeling Language (UML) for designing flexible and distributed enterprise systems. MDA enhances software interoperability, reusability, portability, maintainability, and reliability. UML-based Class Diagrams are used to capture system structure and behavior, facilitating the generation of models and code, including Web Services Description Language (WSDL) for web services. The paper proposes a seamless transformation method and algorithm to convert UML Class Diagrams into WSDL representations and vice versa. This transformation process effectively bridges the gap between modeling and implementation, enabling accurate and efficient translation of design artifacts into executable web services. The proposed approach contributes to the seamless integration of design and implementation phases, promoting consistency, reusability, and interoperability in web service architectures. In summary, this research provides insights into how MDA, UML, and WSDL can work together to design flexible and reliable enterprise systems.

KEYWORDS

Model Driven Approach, Unified Modeling Language, Web Service Description Language, Class Diagrams, Modeling

1. INTRODUCTION

This paper presents a custom C# tool that employs a model-driven approach to enhance the integration and interoperability of web services. The tool provides efficient transformation methods and algorithms for converting class diagrams to WSDL representations and vice versa. While web services are essential for communication and data exchange between diverse systems, manually bridging the gap between object-oriented models and service-oriented architectures through class diagram to WSDL transformation can be challenging and time-consuming[2].

To address this challenge, our tool adopts a model-driven approach and incorporates transformation methods for class to WSDL and WSDL to class diagram conversions. These automated transformations streamline the development and integration of web services. The class to WSDL transformation allows developers to efficiently expose the functionalities of existing classes as web services, while the WSDL to class diagram transformation generates comprehensive class diagrams from existing WSDL descriptions, facilitating the smooth integration of external web services.

To validate our tool's practicality and effectiveness, we conduct a case study on a library service. Through the class to WSDL transformation, we demonstrate how the

library service can be exposed as a web service with well-defined operations and data types. Similarly, the WSDL to class diagram transformation showcases the generation of precise class diagrams for seamless incorporation of the library service into other applications.

By introducing a tool with class to WSDL and WSDL to class diagram transformation methods, we aim to empower developers with a comprehensive solution to enhance the efficiency and effectiveness of web service integration and interoperability. Our model-driven approach automates the transformation processes, reducing manual effort and ensuring consistency in web service development and maintenance.

2. AIM

The aim of this research is to develop a modeling tool based on the Model-Driven Approach that effectively addresses the challenge of maintaining both forward and backward engineering for existing web services.

3. RELATED WORK

The web service environment is complex and dynamic, and Model-Driven Architecture (MDA) is an emerging technology that is still developing. While MDA promises complete code generation, the languages and tools are not yet advanced enough to achieve this. Manual changes are often needed in the generated code, although tools can still have a significant impact on software development. To effectively use MDA, there should be a focus on developing complete and consistent models. Some researchers have explored issues in web service evolution, but few have considered the MDA approach.[3]

In 2003, Marcos proposed an innovative approach using UML to model web services, specifically focusing on the Web Services Description Language (WSDL). The approach demonstrated the transformation of UML models into WSDL descriptions using UML extensions and the tool Rational Rose. [5]

In 2005, Marcos and colleagues further advanced the automatic generation of WSDL documents from UML models within the context of MDA. They compared different transformation tools and showcased their methodology's potential through a successful case study. [6]

In 2006, another group proposed a medical information system that incorporated MDA and web services. They utilized a WSDL metamodel, class diagrams, and the ATL transformation language for conversion. They also introduced three rules for the transformation process.[7] In the same year, another group proposed a transformation methodology for models to service description based on MDA. They developed a mapping algorithm establishing a correspondence between UML and XML Schema elements, demonstrating the efficacy of their approach through an illustrative example. [2]

In 2010, a top-down approach for web service development was introduced, involving analyzing requirements, designing WSDL meta-models using class diagrams, and generating WSDL files. A case study in the e-Health domain showcased the benefits of utilizing modeling and code generation tools. The paper served as a practical guide for efficient web service development with a focus on interoperability and integration. [1]

While class diagrams are commonly used for web service descriptions, they may not provide a complete mapping to WSDL documents. Therefore, the use of UML extensions is necessary. The passage proposes a UML extension that incorporates the WSDL metamodel to enable the generation of WSDL documents from class diagrams. The researchers introduce a novel method and algorithm for transforming class diagrams to WSDL documents and vice versa, along with the development of a modeling tool based on this approach.

4. MODEL DRIVEN APPROACH AND CODE DRIVEN APPROACH

The Model-Driven Approach (MDA), also called the Top-down Approach, and the

Code-Driven Approach, also known as the Bottom-up Approach, are the two main approaches for developing web services. Understanding the differences between these approaches is essential for selecting the best strategy for web service development, as each has its own benefits and drawbacks.

4.1 Model Driven Approach (Top-Down development)

The Model Driven Approach (MDA) is a software development methodology that emphasizes using models as the engine for creation. It separates system specifications from implementation specifics, allowing developers to operate at a higher degree of abstraction. MDA involves developing high-level, platform-independent models (PIMs) and platform-specific models (PSMs) to capture system requirements and design. This top-down development process allows for increased productivity, reusability, and maintainability.[1]

MDA encourages consistency within the system, making it more reliable and reducing errors. However, creating accurate and thorough models can be challenging and time-consuming for massive systems[2]. The availability of proven and trustworthy model transformation tools is crucial for MDA's effectiveness.

To address these challenges, researchers and practitioners have been actively working on advancing model transformation techniques. Several model transformation languages and tools have emerged, simplifying and automating the process of transforming models between different levels of abstraction[8]. These tools not only improve the efficiency of model-driven development but also enhance the accuracy and completeness of generated models.

Furthermore, MDA's benefits extend beyond initial development. By fostering a separation between system specifications and implementation details, MDA allows for more straightforward system maintenance and updates. When a change is required, developers can modify the higher-level models, and the transformations take care of updating the lower-level implementation details accordingly. This capability greatly reduces the risk of introducing errors during maintenance and enhances the system's overall stability.

In conclusion, the Model Driven Approach offers a powerful methodology for software development through its emphasis on using models to drive the creation process. Despite the challenges of creating detailed models for complex systems, the availability of reliable model transformation tools and languages helps maximize the benefits of MDA. Embracing MDA can lead to more robust, consistent, and maintainable software systems, making it an essential approach for modern software engineering practices.

4.2 Code-Driven Approach(Top-down Approach)

In contrast to the top-down nature of the Model Driven Approach, the Code-Driven Approach is a bottom-up development methodology that focuses on writing code first and then generating service contracts like WSDL from that code. This approach is particularly useful when existing codebases need to be exposed as web services or when developers are already familiar with a specific programming language or technology.[8]

In the Code-Driven Approach, developers start by writing the code that defines the functionality of the web service. Once the code is implemented, tools or frameworks can be used to automatically generate the corresponding service contract, such as a WSDL document. This contract provides a standardized way for other systems to communicate with the web service.[8]

The Code-Driven Approach offers benefits in scenarios where developers are more comfortable with coding and prefer to have direct control over the implementation details. It allows for more flexibility in the design and implementation of the web service, as developers can fine-tune the service contract based on the specific requirements of the system.

However, the Code-Driven Approach may also have some drawbacks[1]. Since the

development process starts with code, there is a risk of not fully considering the overall system design before implementation. This can lead to a less coherent and less maintainable system in the long run. Additionally, manually creating service contracts from code can be error-prone and time-consuming, especially for larger projects.

5. PROPOSED METHOD AND ALGORIGHM

Service-Oriented Architecture (SOA) breaks down software systems into smaller, independent services that communicate through well-defined interfaces. Web services, MOM, and ESB are technologies based on SOA principles. Web services use XML, SOAP, and WSDL for distributed system communication. In software development, the top-down approach (Model Driven Approach) emphasizes modeling and generates code based on a service contract (WSDL), while the bottom-up approach (Code-Driven Approach) creates code first and generates the service contract.[4] Our research proposes an MDA method for web service development using UML class diagrams, offering a systematic way to develop web services based on MDA principles.

5.1 Structure of WSDL

The success of Web Services can be attributed to their effective method of describing communication protocols and message formats in a structured manner. WSDL plays a crucial role in separating abstract information from system details[5][6], such as deployment rules and data structures. This separation allows for the reuse of platform-independent information in various forms, including Type, Message, PortType, Binding, and Service. To provide a more detailed overview, Table 1 showcases the specific information encompassed in WSDL, highlighting the inclusion of system details. By employing this approach, Web Services enable interoperability and flexibility, empowering developers to focus on core functionality while ensuring compatibility across diverse platforms and systems. [2]

Table 1. Elements used by WSDL in describing Web Service [2]

Form	Definitions &Functions
Type	Providing data type definitions for message exchange description.
Message	Representing the abstract definition of the transferring data. Message is formed with a set of logic fragment, each one of which correlates with specific Type.
PortType	Referring to the collection of abstract operations. Every abstract operation imports one input message and one output message.
Binding	The binding of PortType to its corresponding specific protocols and data types. Binding rules can be reused in the designing.
Port	Defined as a protocol-datatype correlation pair with a set of addresses for Web Services access. One Port is a single terminal point for access.
Service	A collection of related service access terminals.

5.2 WSDL Metamodel in Class Diagram

In this section, we will discuss how to represent WSDL as a class diagram. According to Table 1 of the WSDL specification, the <definition> element acts as the root element for <types>, <message>, <portType>, <binding>, and <service> elements[5][6]. Based on this, we can create a class diagram, as depicted in figure 1, that captures the structure of WSDL. The WSDL class diagram consists of essential classes that define the structure and behavior of a web service. The "Definition" class serves as

the root element, providing properties like "targetNamespace" and "default namespace" to uniquely identify elements and types and avoid naming conflicts. The "element," "complexType," and "simpleType" classes define and describe data types, ensuring proper formatting and validation. The "Message" class defines the format and structure of exchanged messages, promoting mutual agreement between the web service and clients. The "PortType" class represents supported operations, enhancing usability and integration, while the "Binding" class establishes the protocol and transport mechanism, facilitating correct invocation. Finally, the "Service" class represents the web service itself, providing endpoint information for client access. Overall, this WSDL class diagram enables effective communication and interaction between the web service and clients by providing a clear representation of its structure and facilitating mutual understanding.

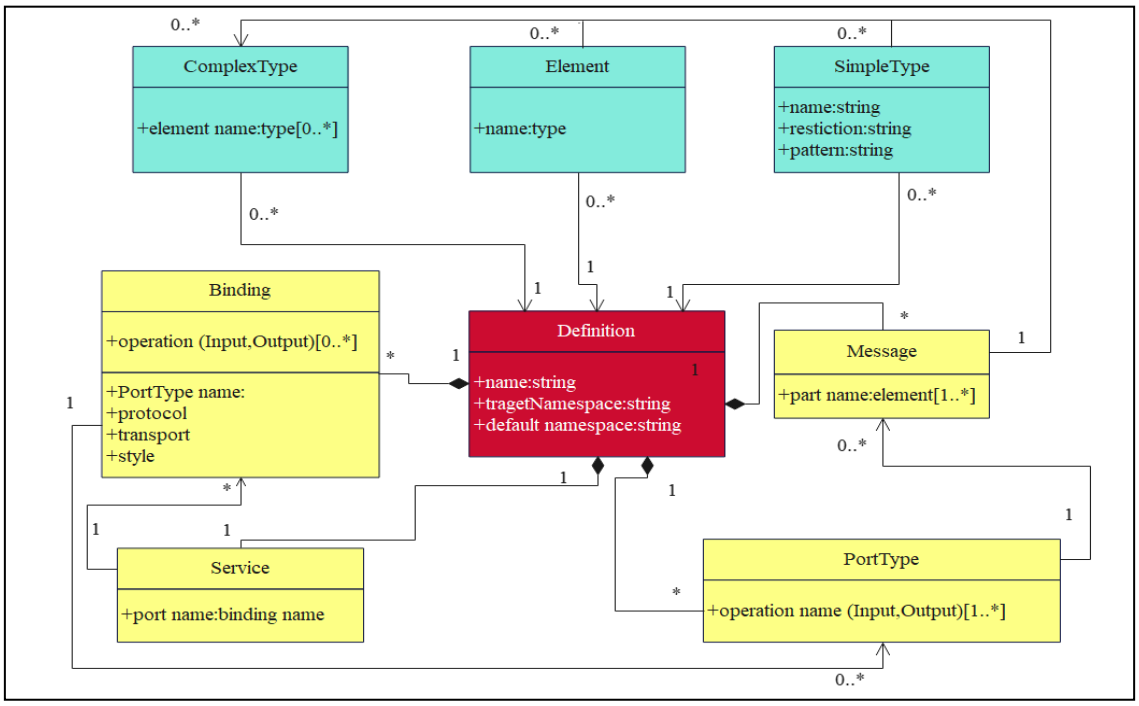


Figure 1. Presentation for WSDL Metamodel with Class Diagram

5.3 Proposed Method for Class to WSDL

This document outlines methods for converting class diagrams to WSDL elements, which are essential for generating WSDL documents from class diagrams. These methods provide a framework for representing classes and their properties as WSDL elements and attributes. The following rules provide guidelines for representing classes and their properties as WSDL elements and attributes:

(a) **Rule 1**: All classes in the class diagram must be represented as elements in the WSDL document.

(b) **Rule 2**: In a class, each property has a name and a value. Rule 2 is defined based on the value of a property. If the value of a property is a primitive type, it should be represented as an attribute. If the value of a property is a collection class ("ListDictionary" or "Dictionary"), it should be represented as an element.

5.4 Proposed Algorithm for Class to WSDL

The Algorithm for Class diagram to WSDL Transformation is a process that takes a class as input and generates a corresponding WSDL document as output. This algorithm is useful for developers who need to expose their classes as web services. The flowchart consists of several major components, including the input (the class), the output (the WSDL document), and several intermediate steps. These steps include checking the stereotype of the class, checking the properties of the class, and generating the WSDL elements.

The class-to-WSDL transformation process, depicted in figure 2, involves two main steps. In the first step, each property of the class in the class diagram is examined, and its corresponding representation in the WSDL document is determined based on rule 2. Primitive types are represented as attributes, collection classes as elements, and inner collection classes as attributes of the outer collection class. This step ensures the accurate representation of all properties in the WSDL document. The second step focuses on generating the actual WSDL elements and attributes using the property values and following transformation guidelines. This step is crucial for creating a complete and precise WSDL document that faithfully represents the class diagram.

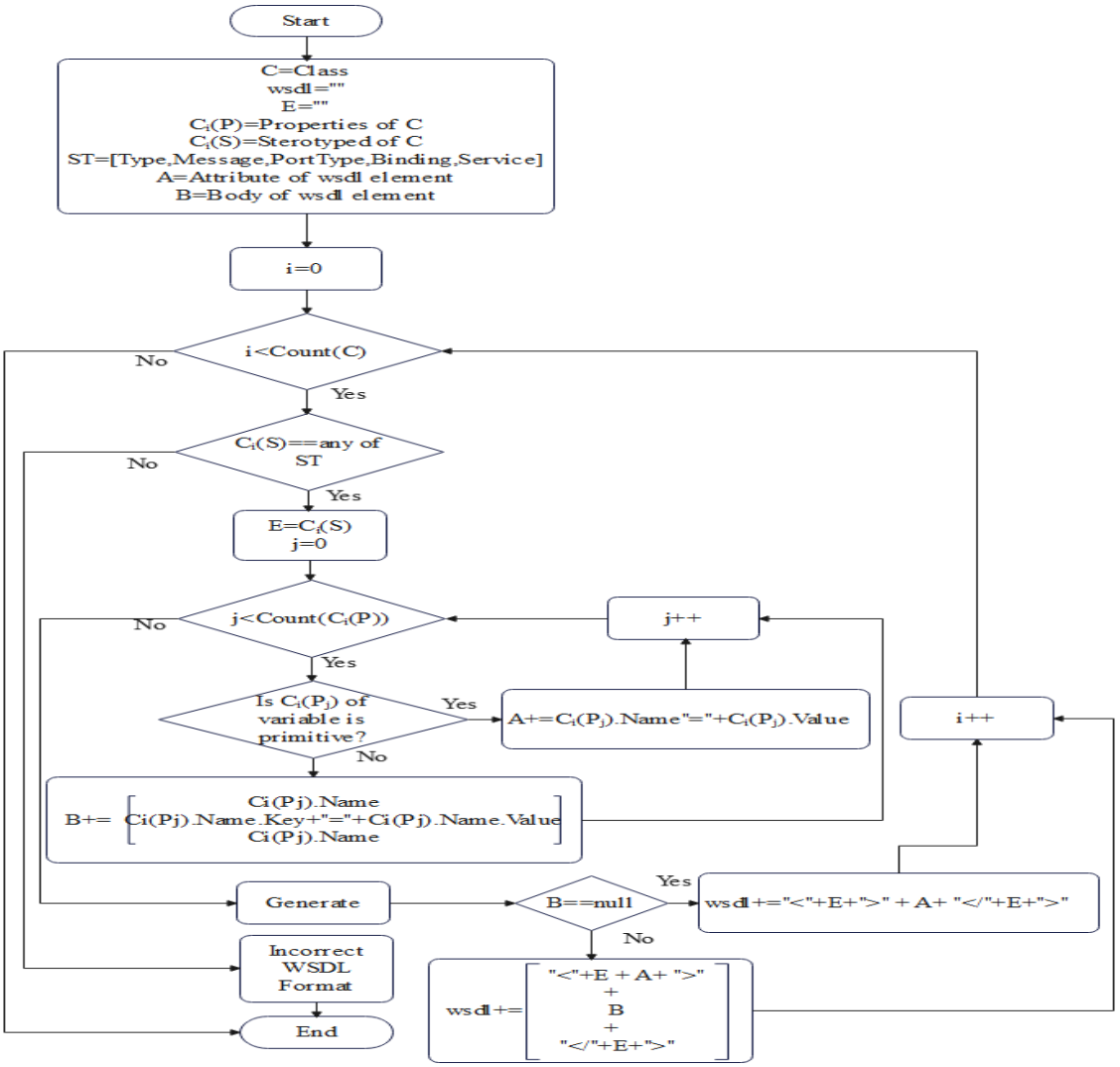


Figure 2. Algorithm for Class to WSDL Transformation

5.5 Proposed Method for WSDL to Class

The transformation of a WSDL document into a Class Diagram is a crucial step in creating a comprehensive service-oriented architecture. The WSDL document specifies the web service interface, including its operations, data types, and message formats, and converting it into a Class Diagram allows for the representation of the service's behaviour and data flow.

(a) Rule 1: All attributes in the WSDL must be represented as properties.

(b) Rule2: If an element has attribute and the element is a child of either the <definition> element or the <schema> element, it must be transformed as a class; otherwise, it must be considered a property.

5.6 Proposed Algorithm for WSDL to Class

Converting a WSDL to a Class Diagram involves creating a class for each WSDL element and attribute. This algorithm is particularly useful for developers who need to modify or update existing WSDL to build web services. The initial stage of algorithm is parsing the incoming WSDL to extract its elements and attributes. These elements and attributes are then transformed into corresponding classes and properties, as shown in figure 3.

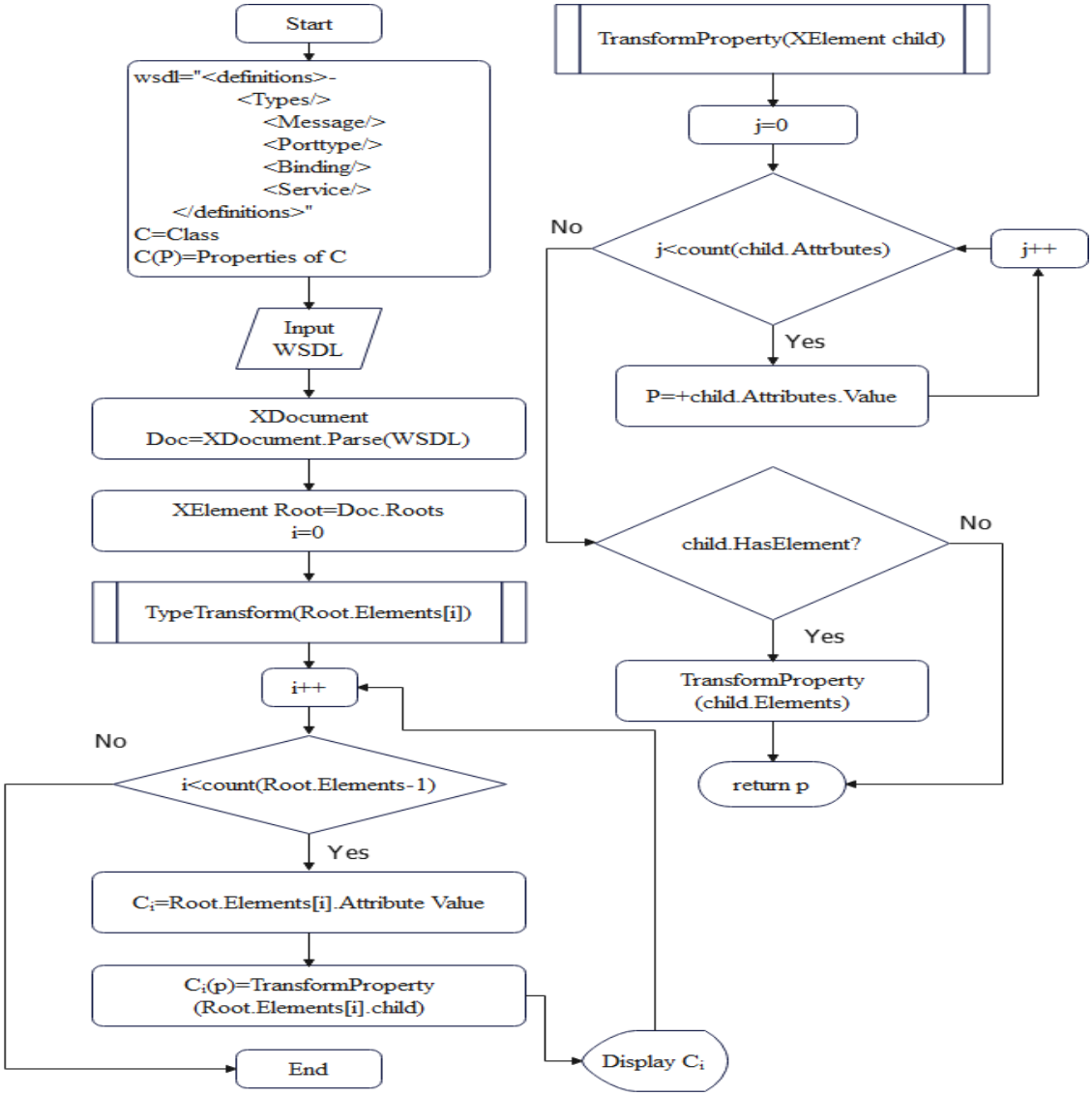


Figure 3. Algorithm for WSDL to Class Transformation

In figure 3, the WSDL transformation process involves obtaining and transforming the incoming WSDL using the "XDocument" functionality. The root element of the "XDocument" represents the <definitions> element in the WSDL. The algorithm proceeds by transforming the child elements of the root element, which include <types>, <message>, <portType>, <binding>, and <service>, into classes of properties, except for the <types> element. The <types> element utilizes XML Schema Document and contains elements such as <element>, <complexType>, and <simpleType>. These elements can be defined in a global style directly under the <schema> element, promoting reusability and interoperability throughout the WSDL document.

The algorithm identifies and assigns the nested elements within the <types> element using XElement, while the global elements directly under the <schema> element are constructed as classes, with their attributes and child elements defined as properties. After handling the <types> element, the algorithm loops through the remaining elements (<message>, <portType>, <binding>, and <service>) and builds them as classes recursively, defining their attributes and child elements as properties. The details of this step are illustrated in figure 3 and figure 4.

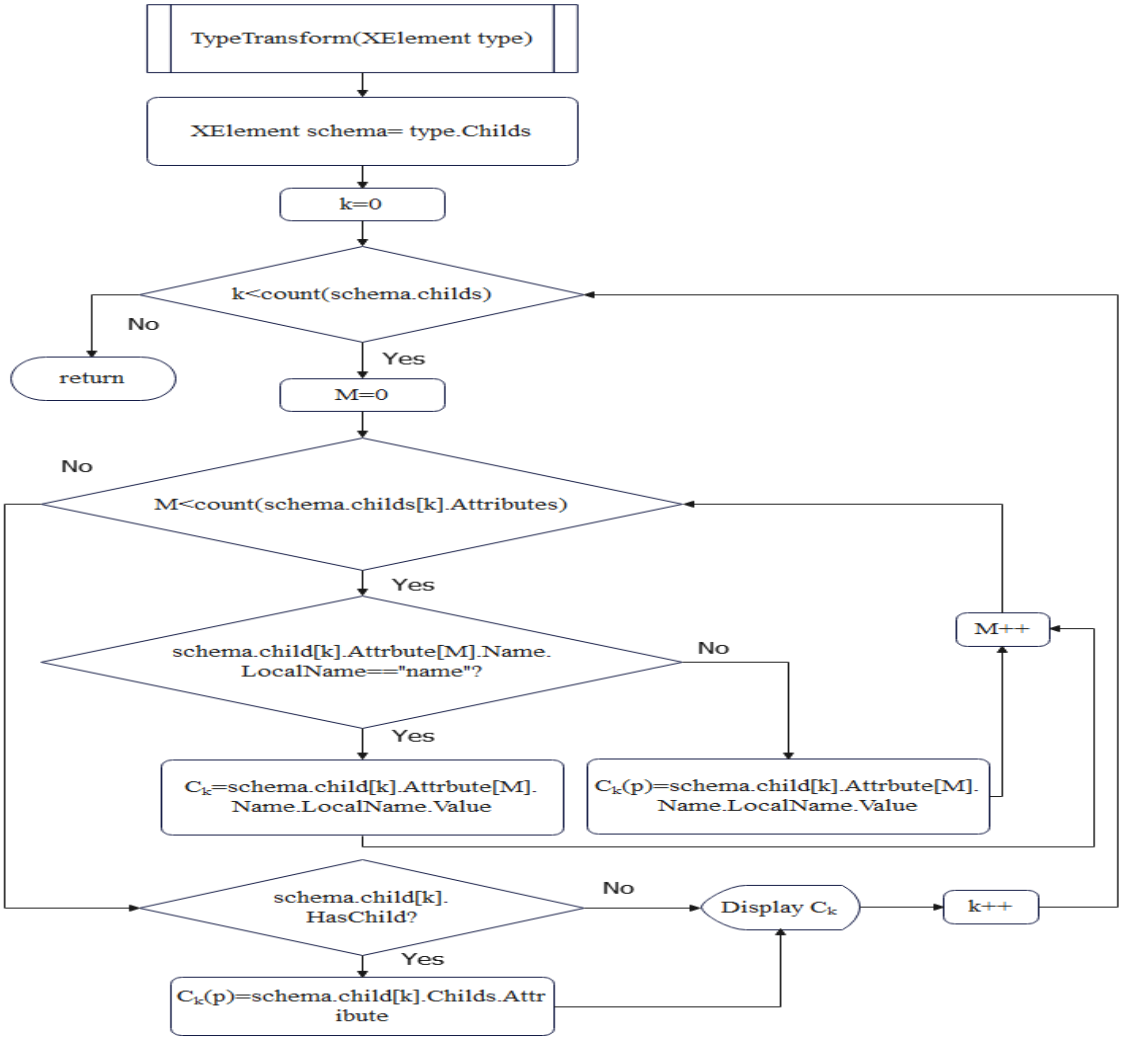


Figure 4. Algorithm for <types> Element to WSDL Transformation

6. CASE STUDY

In this section, we will use the example of the "Library" service, which is an existing WSDL, to demonstrate how to compose and modify the WSDL by adding a new method. By examining the structure and components of the existing WSDL, we can understand the necessary steps to extend its functionality.

6.1 Parsing WSDL into Modeling Tool

Based on our WSDL to Class transformation method and algorithm, we create incoming wsdl as associated class diagram. The provided WSDL already includes two operations: "addBook" and "getBookById" methods. These operations define the functionality available in the "Library" service. The "addBook" method allows for adding new books to the library, while the "getBookById" method retrieves a book from the library based on its unique identifier. These existing operations provide specific functionality for interacting with the "Library" service. The original WSDL format is as follows:

```
<wsdl:portType
name="library_service">
  <wsdl:operation name="addBook">
   <wsdl:input
message="tns:addBookReq" />
   <wsdl:output message= "tns:
addBookRsp" />
 </wsdl:operation>
<wsdl:operation
name="getBookById">
  <wsdl:input
message="tns:getBookByIdReq" />
    <wsdl:output-
message="tns:getBookByIdRsp"/>
</wsdl:operation>
</wsdl:portType>
```

After parsing the above WSDL, the associated class diagram is shown in figure 5.

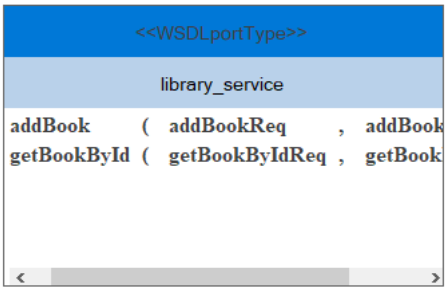


Figure 5. Class Model for <portType>

The class model in figure 5 represents the <portType> element in the WSDL. It visualizes the structure and relationships of the operations within the "Library" service. The "addBook" operation is represented by a method with corresponding input and output messages, while the "getBookById" operation follows the same pattern. This class model provides a clear understanding of the existing functionality of the "Library" service as defined in the WSDL.

By analyzing the parsed WSDL and its corresponding class model, we can proceed with extending the functionality of the "Library" service by adding a new method. The next section will outline the steps and modifications required to achieve this extension.

6.2 Extending the Functionality: Adding the "deletebook" Method

To extend the functionality of the "Library" service by adding the "deletebook" method, we need to make the necessary modifications to the class diagrams such as “Type”, “Message”, “PortType”, “Binding” and “Service” classes. Figure 6 shows the modification of “PortType” class.

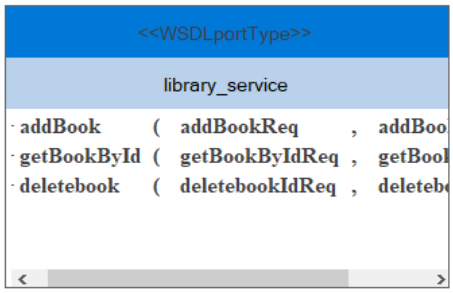


Figure 6. Updated Class Model for <portType>

Based on the modifications made in the class diagram, specifically in the "PortType"

class, we have added the "deletebook" method to extend the functionality of the "Library" service. The updated WSDL document reflecting these modifications is as follows:

```
<wsdl:portType
name="library_service">
  <wsdl:operation name="addBook">
   <wsdl:input
message="tns:addBookReq" />
   <wsdl:output message=
"tns:addBookRsp"/>
 </wsdl:operation>
 <wsdl:operation
name="getBookById">
  <wsdl:input
message="tns:getBookByIdReq" />
    <wsdl:output--
message="tns:getBookByIdRsp"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:operation name="deletebook">
  <wsdl:input
message="tns:deletebookIdReq" />
    <wsdl:output-
message="tns:deletebookIdRsp"/>
</wsdl:operation>
<</wsdl:portType>
```

In the updated WSDL, the "deletebook" method has been added as a new operation under the "library_service" portType. It has an input message of "tns:deletebookIdReq" and an output message of "tns:deletebookIdRsp". This addition reflects the extension of the "Library" service to include the functionality of deleting a book.

By following the Class to WSDL transformation method and algorithm, the modified class diagrams were used to generate the updated WSDL document. This ensures that the WSDL accurately represents the extended functionality, including the newly added "deletebook" method, in the "Library" service.

6.3 Validating the Generated WSDL in NetBeans

The implementation of the "Library" service in NetBeans using the generated updated WSDL involves importing the updated WSDL document into the NetBeans IDE. This imported WSDL reflects any modifications made to the service. By doing so, developers can automatically generate the necessary code and configurations for the service implementation.

In the case of the "Library" service, NetBeans generates an abstract class called "Library _service" that contains methods corresponding to the operations defined in the WSDL that codes are shown in below:

```
public class Library_service
{
public boolean addBook(java.lang.String
arg1)
{
   //TODO implement this method
    throw new
UnsupportedOperationException
    ("Not implemented yet.");
}
public java.lang.Object getBookById(int
arg2)
{
  //TODO implement this method
  throw new
UnsupportedOperationException
   ("Not implemented yet.");
}
public boolean deletebook(int deleteId)
{
  //TODO implement this method
  throw new
UnsupportedOperationException
  ("Not implemented yet.");
}
}
```

These methods include "addBook," "getBookById," and the newly added "deleteBook." However, the implementation of these methods is not provided and needs to be completed by the developer. By implementing the necessary code within the generated abstract class, developers can fulfill the functionality required by each operation. This approach saves time and effort as developers don't have to manually write boilerplate code or re-implement the service from scratch. Instead, they can focus on customizing and adapting the code to suit the specific requirements of the service.

It's important to note that the provided code snippet includes placeholders ("//TODO implement this method") indicating where the developer should add their implementation logic. These placeholders are a reminder to complete the implementation and remove the “UnsupportedOperationException”, which is a default exception thrown when the method is called but not implemented.

6.4 Comparison of the New and Old Method

In this section, the paper introduces a case study involving the "Library" service to demonstrate how to extend its functionality by adding a new method. The section begins with parsing the existing WSDL and creating an associated class diagram to visualize the structure and relationships of the operations within the "Library" service. The two existing methods, "addBook" and "getBookById," are analyzed and understood as they define the current functionality available in the service.

The paper then proceeds to extend the functionality by introducing the "deletebook" method. The modifications required for the class diagrams, such as "PortType," are demonstrated in Figure 6, which reflects the addition of the new method to the "Library" service. The updated WSDL document is provided to show the changes made to include the "deletebook" operation under the "library_service" portType.

Comparing the old and new methods, it is evident that the original WSDL only had two operations, "addBook" and "getBookById," which provided limited functionality for interacting with the "Library" service. However, with the extension of the "deletebook" method, the service's capabilities are expanded, allowing users to delete books from the library.

Furthermore, the paper explains the process of validating the generated WSDL in NetBeans by importing the updated document. The IDE generates an abstract class, "Library_service," containing methods corresponding to the operations defined in the WSDL, including the newly added "deleteBook" method. It is clarified that developers need to implement the logic within these methods to fulfill the functionality required by each operation.

In summary, the comparison between the old and new methods highlights the evolution of the "Library" service. The addition of the "deletebook" method extends the service's functionality and enables users to perform a broader range of actions. The use of NetBeans IDE streamlines the implementation process by generating the initial code structure, which developers can then customize to suit the specific requirements of the service. Overall, this section effectively demonstrates the process of extending web service functionality and provides a clear understanding of the steps involved in modifying the WSDL to accommodate the new method.

7 CONCLUSION

This paper builds upon previous research on Model Driven Approaches and proposes the novel methods and algorithms for forward and backward engineering in web service development. A case study of an existing "Library" service is presented to demonstrate the practical application of the proposed approach. The study involves adding the existing WSDL document of “Library” service to our modeling tool, describing and modifying it in Class diagrams, regenerating new WSDL document.

To showcase the implementation process, the paper includes the abstract code obtained by integrating the output WSDL file into the NetBeans IDE. By customizing the necessary sections of the code, developers can save time and adopt a top-down approach to their web service development. By combining the proposed transforming method and algorithm with the modeling tool, developers can streamline the forward and backward engineering processes. This approach allows for efficient customization of web service development, reducing manual effort and ensuring a more

systematic and organized development workflow.

In summary, this paper contributes to the field of web service development by presenting a practical and efficient Model Driven Approach for forward and backward engineering. The case study of the "Library" service demonstrates the effectiveness of the proposed method, and the generated abstract code in IDE offers guidance to developers seeking to implement their own projects.

ACKNOWLEDGMENT

I gratefully thank my supervisors Dr. Phone Naing (Professor at Department of Computer Science, Higher Education Center), Dr. Aung Aung Hein (Lecturers at Department of Computer Technology, Higher Education Center) for valuable advices, kind guidance, supports and cooperation. Their energetic and enthusiastic supports are valuable encouragements during a period of study for this paper. Last but not least, many thanks to all person who directly and indirectly contributed toward the success of this paper.

REFERENCES

[1] Alexandre Bellini, Antonio Franciso do Prado, Luciana Aparecida and Martinex Zania, “Top-Down Approach for Web Services Development”, Fifth International Conference on Internet and Web Applications and Services, pp.426-431,2010.

[2] Bin Yu, Chen Zhang, and Yang Zhao, “Transform Model to Service Description Based on MDA”, IEEE Asia-Pacific Conference on Services Computing, pp.605-608,2006.

[3] Božić, Velibor., “Model Driven Architecture (MDA)-The Way to Develop Systems in the Future”, ResearchGate, February 2023. DOI:10.13140/RG.2.2.33116.87683

[4] Claus T. Jensen, “SOA Design and Principle for Dummies”, IBM, 2013.

[5] Marcos, E., de Castro, V., Vela, B, “Representing Web Services with UML: A Case Study”, Springer, Lecture Notes in Computer Science, Volume 2910, pp.17-27, 2003, Heidelberg, Berlin.

[6] Marcos, E., de Castro, V., Vela, B, “WSDL Automatic Generation from UML Models in an MDA framework”, International Conference on Next Generation Web Services Practices,pp.6, August 2005.

[7] Simone A. B. Melo, Denivaldo Lopes, and Zair Abdelouahab, “Developing Medical Information System with MDA and Web services”, Proceedings of the International Conference on Software Engineering Research and Practice & Conference on Programming Languages and Compilers (SERP), Volume 2, pp.562-568, June 2006, Las Vegas, USA.

[8] Johnson, A., & Williams, B. (2022). Challenges in Model Driven Software Development. Journal of Software Engineering, 10(2), 45-56.

Authors

Biography

Wai Phyo Maung received M.C.Sc. degree from Higher Education Center (HEC), in 2016. Now he is a Ph.D. candidate of Computer Science Department, (HEC).

Dr. Phone Naing earned his Master’s and Ph.D degree from National Research Nuclear University (MEPhI), (Russian Federation) in 2004 and 2007. Now he is a Professor of Department of Computer Science in HEC.

Dr. Aung Aung Hein earned his master degree Ph.D. degree from National Research Nuclear University (MEPhI), (Russian Federation) in 2009 and 2013. Now he is a lecturer of Department of Computer Technology.

SOLVING PROBLEMS OF SIMPSON'S RULE IN NUMERICAL BY MATLAB PROGRAMMING

Moe Moe Thein[1] and Thandar Lwin [2]

[1,2] Department of Engineering Mathematics, Technological University (Mandalay)
moemoetheinmdy227@gmail.com,thanderlwinthanderlwin@gmail.com

ABSTRACT

The main goals of this paper are to introduce concepts of numerical methods in mathematics and introduce MATLAB in an Engineering conceptual. MATLAB is one of the most popular software packages available today. By using this technological software, we can also solve Simpson's rule in numerical problems. According to this paper, students can find it much more incentive and intellect when MATAB programming is made an integrates parts of mathematics. In this paper, we calculate functions such as exponential function and trigonometric function how to solve Simpson's rule by MATLAB programming. This paper aims at teaching how to program the numerical integration with a step by step approach in transforming their algorithms to the most basic lines of code that can run on the computer efficiently and output the solution at the required degree of accuracy. In addition, the paper also includes a brief overview of these MATLAB commands and program various Simpson's rule problems.

KEYWORDS

Simpson's Rule, Numerical, MATLAB Programming, Integration

1. INTRODUCTION

Instruction on using MATLAB is scattered through the substance on numerical methods. [7] MATLAB programming provides a essential introduction to analysis suitable for undergraduate students in mathematics, computer science, and engineering. Numerical derivation for Simpson's rule is given that it uses elementary results and builds the student's understanding of calculus by MATLAB programming. Computer assignment using MATLAB give student's an opportunity to practice their skills at scientific programming [3]. We will consider for the numerical approximation of definite integrals is known as Simpson's rule. [5] The aim at all times is to use the experience of numerical experiment and a feel for the mathematics to apply numerical methods efficiently. MATLAB provides an enormous range of ready to use programs [1]. It can also serve as a reference to MATLAB applications for professional engineers and scientists [9]. There are many various software packaged available. MATLAB has happened the industry standard for apply in many fields containing mathematics. It's important to understand the different capabilities of each as one software package may be the best tool depending on the task. It is an excellent tool for working with large data sets. [8] Computing today becomes so much more powerful when combined with programming [6]. MATLAB programs are clean and easy to read; they lack the syntactic clutter of some mainstream languages [2]. When MATLAB programming the numerical integration formula, it becomes evident that it works for all mathematical function [6].

2.BACKGROUND THEORY

MATLAB is a high-level software package with many built in functions that make the learning of numerical methods much easier and more interesting [9]. MATLAB, the product of Mathworks Company, is the general-purpose computing software.

The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational

mathematics. MATLAB provides specialized collections of programs applicable to particular problem areas in what are known as toolboxes and which represent the collaborative efforts of top researchers from all over the world [1]. A teacher using technology to motivate students is more powerful and productive than one simply using lectures and textbooks. The used of MATLAB enhanced students' motivation for learning mathematics.

2.1. Numeric Integration

The Numeric integration means the numeric evaluation of integrals $J = \int_a^b f(x)\,dx$ where *a* and *b* are given and *f* is a function given analytically by a formula or empirically by a table of values. Geometrically, *J* is the area under the curve of *f* between *a* and *b* (Fig. 1). We know that if *f* is such that we can find a differentiable function *F* whose derivative is *f*, then we can evaluate *J* by applying the familiar formula

$$J = \int_a^b f(x)\,dx = F(b) - F(a)$$

$[F'(x) = f(x)]$.

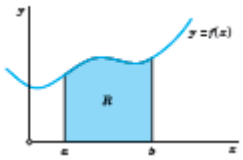


Figure 1. Geometric interpretation of a definite integral [4]

2.2. Simpson's Rule of Integration

To derive Simpson's rule, we divide the integration $a \le x \ll b$ into an even number of equal subintervals, say, n=2m into subintervals of length $h = (ba)/(2m)$ with endpoints $x_0(= a), x_1, \dots, x_{2m-1}, x_{2m}(= b)$; see Fig.2. We now take the first two subintervals and approximate f(x) in the interval $x_0 \ll x \ll x_2 = x_0 + 2h$ by the Lagrange polynomial $p_2(x)$ through $(x_0, f_0), (x_1, f_1), (x_2, f_2), where\ f_j = f(x_j)$.

$$p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f_2. \quad (1)$$

The denominators in (1) are $2h^2, -h^2, and\ 2h^2$, respectively. Setting $s = \frac{(x-x_1)}{h}$,

$x - x_1 = sh, x - x_0 = x - (x_1 - h) = (s+1)h$

$x - x_2 = x - (x_1 + h) = (s-1)h$

And we obtain

$$p_2(x) = \frac{1}{2}s(s-1)f_0 - (s+1)(s-1)f_1 + \frac{1}{2}(s+1)sf_2$$

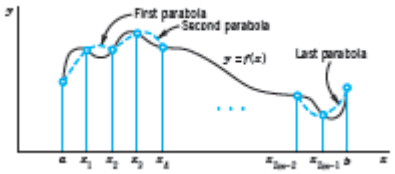


Figure 2. Simpson's rule [4]

We now integrate with respect to $x_0\ to\ x_2$. This corresponds to integrating with respect to s from -1 to 1. Since $dx = h\,ds$, the result is

$$\int_{x_0}^{x_2} f(x)\,dx \approx \int_{x_0}^{x_2} p_2(x)\,dx = h\left(\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{1}{3}f_2\right). \quad (2)$$

$$\int_b^a f(x)dx \approx \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{2m-2} + 4f_{2m-1} + f_{2m}), \quad (3)$$

where $h = (b-a)/(2m)$ and $f_j = f(x_j)$.

Table 1. Simpson's Rule of Integration

Algorithm Simpson (a,b,m,$f_0, f_1, \dots, f_{2m}$)

This algorithm computes the integral $J = \int_a^b f(x)\,dx$from given values $f_j = f(x_j)$ at equidistant $x_0 = a, x_1 = x_0 + h, \dots, x_{2m} = x_0 + 2mh = b$ by Simpson's rule, where $h = \frac{(b-a)}{(2m)}$.

INPUT: $a, b, m, f_0, \dots, f_{2m}$

OUTPUT: Approximate value $\check{J}$ of J

Compute $s_0 = f_0 + f_{2m}$

$s_1 = f_1 + f_3 + \cdots + f_{2m-1}$

$s_2 = f_2 + f_4 + \cdots + f_{2m-2}$

$h = \frac{(b-a)}{2m}$

$\check{J} = \frac{h}{3}(s_0 + 4s_1 + 2s_2)$

OUTPUT $\check{J}$. Stop.

End SIMPSON

2.3. SOLVING PROBLEMS BY MATLAB PROGRAMING

Evaluate $f(x) = \int_0^1 e^{-x^3} dx$ by Simpson's rule

$n = 10$.

Solution:

$$f(x) = \int_0^1 e^{-x^3} dx, 2m = n = 10,$$

$$a = 0, b = 1, h = \frac{1}{10} = 0.1$$

j	x_j	$f(x_j) = e^{-x_j^3}$
0	1	1.0000
1	0.1	0.9990
2	0.2	0.9920
3	0.3	0.9734
4	0.4	0.938
5	0.5	0.8825
6	0.6	0.8057
7	0.7	0.7096
8	0.8	0.5993
9	0.9	0.4824
10	1	0.3679
Sums	1.3679	4.0469 3.3350

By Simpson's Rule,

$$f(x) = \int_a^b f(x)\,dx$$

$$\approx \frac{h}{3}[(f_0 + f_n) + 4(f_1 + f_3 + \cdots + f_{n-1}) + 2(f_2 + f_4 + \cdots + f_{n-2})]$$

$$\int_0^1 e^{-x^3} dx \approx \frac{0.1}{3}[(1.3679) + 4(4.0469) + 2(3.3350)]$$

$$\approx 0.8075$$

MATLAB Function of Simpson's Rule

```
function simpson(f,a,b,n)
h=(b-a)/n;
s0=0;s1=0;s2=0;
Names={'i','x(i)','f(xi)'};
fprintf('\t%s\t\t%s\t\t%s\t\t%s\n',Names{:})
s0=f(a)+f(b);
for i=1:n-1
   x=a+i*h;

fprintf(1,'\n\t%.4f\t\t%4.4f\t\t%8.4f\n',i,x,f(x));
   if mod(i,2)~=0
      s1=s1+f(x);
   else
      s2=s2+f(x);
   end
end
Area=h*(s0+4*s1+2*s2)/3;
fprintf('\nArea=h*(s0+4*s1+2*s2)/3=%8.4f\n',
Area)
```

When we run the MATLAB function of Simpson's Rule problem, the output is

```
>> f=@(x)exp(-x^3)
   f =   @(x)exp(-x^3)
>> a=0; b=1; n=10;
>> h=1/10
h =
   0.1000
>> simpson (f, a, b, n)
  i          x(i)          f(xi)
  1.0000     0.1000        0.9990
  2.0000     0.2000        0.9920
  3.0000     0.3000        0.9734
  4.0000     0.4000        0.9380
  5.0000     0.5000        0.8825
  6.0000     0.6000        0.8057
  7.0000     0.7000        0.7096
  8.0000     0.8000        0.5993
  9.0000     0.9000        0.4824
Area=h*(s0+4*s1+2*s2)/3=   0.8075
```

Evaluate $f(x) = \int_1^2 \frac{2}{x} dx$ by Simpson's rule $n = 10$.
Solution:

$$f(x) = \int_1^2 \frac{2}{x} dx, a = 1, b = 2, n = 10, h = 0.1$$

j	x_j	$f(x_j) = \frac{2}{x_j}$	
0	1	2.0000	
1	1.1		1.8182
2	1.2		1.6667
3	1.3		1.5385
4	1.4		1.4286
5	1.5		1.3333
6	1.6		1.2500
7	1.7		1.1765
8	1.8		1.111
9	1.9		1.0526
10	2	1.0000	
Sums	3.0000	6.9191	5.4564

By Simpson's Rule,

$$f(x) = \int_a^b f(x)\,dx$$

$$\approx \frac{h}{3}[(f_0 + f_n) + 4(f_1 + f_3 + \cdots + f_{n-1}) + 2(f_2 + f_4 + \cdots + f_{n-2})]$$

$$\int_1^2 \frac{2}{x} dx \approx \frac{0.1}{3}[3 + 4(6.9191) + 2(5.4564)] \approx 1.3863$$

When we run the MATLAB function of Simpson's rule problem, the output is

```
>> f=@(x)2/x
f = @(x)2/x
>> a=1;b=2;n=10;
>> h=1/10
h =0.1000
>> simpson(f,a,b,n)
```

i	x(i)	f(xi)
1.0000	1.1000	1.8182
2.0000	1.2000	1.6667
3.0000	1.3000	1.5385
4.0000	1.4000	1.4286
5.0000	1.5000	1.3333
6.0000	1.6000	1.2500
7.0000	1.7000	1.1765
8.0000	1.8000	1.1111
9.0000	1.9000	1.0526

Area=h*(s0+4*s1+2*s2)/3= 1.3863

Evaluate $f(x) = \int_0^{1.25} sinx^3 dx$ by Simpson's rule $n = 10$.
Solution:

$f(x) = \int_0^{1.25} sinx^3 dx$,$a = 0, b = 1.25, n = 10$

$h = \frac{1.25}{10} = 0.125$

j	x_j	$f(x_j) = \sin(x_j^3)$		
0	0	0		
1	0.1250		0.0020	
2	0.2500			0.0156
3	0.3750		0.0527	
4	0.50000			0.1247
5	0.6250		0.2417	
6	0.7500			0.4095
7	0.8750		0.6209	
8	1.0000			0.8415
9	1.1250		0 .9892	
10	1.2500	0.9278		
Sums		0.9278	1.9065	1.3913

By Simpson's Rule,

$$f(x) = \int_a^b f(x)\,dx$$
$$\approx \frac{h}{3}[(f_0 + f_n) + 4(f_1 + f_3 + \cdots + f_{n-1}) + 2(f_2 + f_4 + \cdots + f_{n-2})]$$
$$\int_0^{1.25} \sin x^3 dx \approx \frac{0.125}{3}[0.9278 + 4(1.9065) + 2(1.3913)$$
$$\approx 0.4724$$

When we run the MATLAB function of Simpson's rule problem, the output is

```
>> f=@(x)sin(x^3)
f = @(x)sin(x^3)
>> a=0; b=1.25; n=10;
>> h=1.25/10
h =0.1250
>> simpson (f, a, b, n)
```

i	x(i)	f(xi)
1.0000	0.1250	0.0020
2.0000	0.2500	0.0156
3.0000	0.3750	0.0527
4.0000	0.5000	0.1247
5.0000	0.6250	0.2417
6.0000	0.7500	0.4095
7.0000	0.8750	0.6209
8.0000	1.0000	0.8415
9.0000	1.1250	0.9892

Area=h*(s0+4*s1+2*s2)/3= 0.4724

Evaluate $f(x) = \int_0^1 \frac{x}{cosx}\,dx$ by Simpson's rule $n = 10$.

Solution:

$f(x) = \int_0^1 \frac{x}{cosx}\,dx$,$a = 0, b = 1, n = 10$

$h = \frac{1}{10} = 0.1$

j	x_j	$f(x_j) = \frac{x_j}{cosx_j}$		
0	0	0		
1	0.1		0.1005	
2	0.2			0.2041
3	0. 3		0.3140	
4	0.4			0.4343
5	0.5		0.5697	
6	0.6			0.7270
7	0.7		0.9152	
8	0.8			1.1483
9	0.9		1.4479	
10	1	1.8508		
Sum		1.8508	3.3473	2.5137

By Simpson's Rule,

$$f(x) = \int_a^b f(x)\,dx$$
$$\approx \frac{h}{3}[(f_0 + f_n) + 4(f_1 + f_3 + \cdots + f_{n-1}) + 2(f_2 + f_4 + \cdots + f_{n-2})]$$

$$f(x) = \int_0^1 \frac{x}{cosx}\,dx$$

$$\approx \frac{0.1}{3}[1.8508 + 4(3.3473) + 2(2.5137)$$
$$\approx 0.6756$$

When we run the MATLAB function of Simpson's rule problem, the output is

```
>> f=@(x) x/cos(x)

f = @(x) x/cos(x)

>> a=0; b=1;n=10;

>> h=1/10;
```

```
>> simpson (f, a, b, n)

 i          x(i)        f(xi)
1.0000     0.1000      0.1005
2.0000     0.2000      0.2041
3.0000     0.3000      0.3140
4.0000     0.4000      0.4343
5.0000     0.5000      0.5697
6.0000     0.6000      0.7270
7.0000     0.7000      0.9152
8.0000     0.8000      1.1483
9.0000     0.9000      1.4479

Area=h*(s0+4*s1+2*s2)/3=   0.6756
```

3. CONCLUSIONS

MATLAB programming is one of the most useful and important teaching tools for mathematics. In this paper, MATLAB programming is used to solved problems of Simponson's in numerical. It broadens student's understanding and enhances numerical integration skills. So, numerical course should be taught in such a way so that the students can be able to develop their problem-solving skills on numerical integration to solve verification problems using MATLAB.

ACKNOWLEDGEMENTS

I greatly thank to Dr. Su Yin Win, Rector, University of Technology (MDY), for his permission to take out this paper. I would like to greatly thank to Dr Hla Hla Myint, Professor and Head, Engineering Mathematics, University of Technology (Mdy) for his kind approval to submit this paper, Dr Aye Thandar Lwin, Associate Professors at Engineering Mathematics for their arrangement and valuable suggestions in completion of paper. Special thanks are due to University of Computer Studies (JITRI) and Technical Program Committee of , for giving a chance submit paper. Very special thanks to Daw May Zin Aye, Associate Professor at English Department for her valuable supports from the language point of view in my paper. Finally, I greatly thank my parents and husband for their encouragement during writing of my paper.

\

REFERENCES

[1] C. Woodford.C. Phillips, "Numerical Methods with Worked Examples: Matlab Edition (second Edition)", Newcastle upon Tyne, UK.

[2] JAAN KIUSALAAS, "Numerical Methods in Engineering with MATLAB SECOND EDITION", Published in the United States of America by Cambridge University Press, New York.

[3] John H. Mathews and Kurtis D. Fink, "Numerical Methods Using MATLAB (Third Edition)", Prentice Hall, Upper Saddle River, NJ 07458.

[4] Kellaway, F.W. (1969). Advance Engineering Mathematics. By Erwin Kreyzig. Pp.xx, 899,685.(Wiley).The Mathematical Gazette 53(386),444-444.

[5] P. Howard, "MATLAB for M152B", Spring 2009.

[6] Svein Linge. Hans Petter Langtangen, "Programming for Computations-MATLAB Octave", Springer Heidelberg Dordrecht London New York, December 2015.

[7] Todd Young and Martin J. Mohlenkamp, "Introduction to Numerical Methods and MATLAB Programming for Engineers", May4,2017.

[8] Troy Siemers, "AN INTRODUCTION TO MATLAB AND MATHCAD", Liscensed to the public under Creative Commons.

[9] WON YOUNG, WENWU CAO, TAE-SANG CHUNG, JOHN MORRIS, JOHN MORRIS, "Applied Numerical Methods Using MATLAB", A JOHN WILEY & SONS, INC., Publication, October 2004.

Authors

Biography

First A. My name is Moe Moe Thein and I am lecturer at Department of Engineering Mathematics from TU (Mandalay). I was born in 1987 and my native town is Mandalay. I got first degree (B.Sc)(Hons)(Maths) at Yadanabon University in 2007. I got master degree (M.Sc)(Maths) at Yadanabon University in 2010. I worked as a tutor at (UTYCC) in 2014.Then I was promoted as an assistant lecturer at (UTYCC) in 2018. I was promoted as a lecturer at TU(Mdy) in 2022.

Biography

Second B. My name is Thandar Lwin and I am associate professor at Department of Engineering Mathematics from TU(Mdy). I was born in 1977 and my native town in Mandalay. I got first degree (B.Sc)(Hons) (Maths) at Mandalay University in 2003. I got master degree (M.Sc) (Maths) at Mandalay University in 2005. I worked as a tutor at TU(Meiktila) in 2005.I was promoted as a assistant lecturer at TU(Meiktila) in 2013. I was promoted as a lecturer at MTU in 2017. I was promoted as an associate professor at TU (Keng Tung) in 2021. I was transferred at TU(Mdy) in 2021.

1.TU(Pakokku)journal: Engineering Research (Volume 01, Issue 02, May, 2020) 15.5.2020 Title. Application and Molding of RLC Circuit with Unit Step Function By Laplace Transform (Page-220 to223) (First author) Technological University (Pakokku) Journal: Engineering and research 2020.

2.MTUJSET (VOLUME6, ISSUE 4, December, 2019) Title, Application of Prim's Minimum Weight Spanning Tree Algorithm for Road Map Estimation in Myanmar Page-148 to152 (First author) Mandalay Technological University Journal of Science, Engineering And Technology (29.11.2019)

A NEW SERVICE BROKER ROUTING POLICY FOR DATACENTER SELECTION IN CLOUD COMPUTING

Kaung Myat Soe[1] and May Phyo Thu[2]

[1,2]Faculty of Computer Science, University of Computer Studies (Mandalay)
kaungmyatsoemdy1@gmail.com, mayphyothu.mpt1@gmail.com

ABSTRACT

Cloud computing provides computing resources as services to cloud users via the Internet. Nowadays, organizations' IT infrastructures are shifting to the cloud. Users will want to process their requests at the lowest possible cost and with the least possible response time, and that directly depends on getting a suitable datacenter. The service broker is responsible for datacenter selection by using service broker routing policies. The existing Service Proximity-Based Routing policy (SPBR) selects the closest datacenter to the user request. However, if there is more than one datacenter in the same region, a random selection is made without considering any factors like response time, datacenter processing time, cost, etc. An improperly selected datacenter will often give unsatisfactory results. Therefore, a new service broker routing policy was proposed to select the suitable datacenter with better processing time and response time. CloudAnalyst simulation framework was used to evaluate the performance of a proposed service broker routing policy. The results show the ability of the proposed policy to improve datacenter selection based on overall response time, and processing time.

KEYWORDS

Cloud Service Broker, CloudAnalyst, Simulation.

1. INTRODUCTION

As almost all applications and IT infrastructure are supported over the Internet, demands for computing resources are shifting from traditional to cloud service providers. According to NIST (National Institute of Standards and Technology) [15], cloud computing is an internet-based model and is a technology in which computing resources delivered to users by service providers over the internet. Cloud computing provides computing resources (e.g. storage, processing power, databases, networking, etc.) as a service to handle various users' demands. These services may include hardware or other software, storage, infrastructure, databases, and many more [28]. There are different types of services provided by cloud service providers, cloud users require an intermediary between them to manage these services and ease the development and deployment of applications. Datacenters are composed of physical machines that have processors, storage devices, and memory. And these datacenters are built in various locations around the world. In particular, in order to effectively use the capacity of datacenters and to ensure the availability of services, datacenters are set up in places closest to users.

But different cloud users have different needs, constraints, and demands. For example, better response time regardless of the cost is a key requirement for someone, but budget may be a constraint for others. If several users have the same demands, the allocated datacenter may encounter overloading and performance degradation. The datacenters' locations, user distributions, the number of datacenters in the region, and the availability of suitable datacenters all have a direct effect on the overall performance in the cloud environment. Therefore, a suitable datacenter selection has become necessary in order to improve overall performance and meet a variety of user demands [29].

Cloud service brokers are something that helps cloud users easily and effectively use the resources supported by cloud service providers.

And cloud service brokers are also responsible for selecting the suitable datacenter. In this regard, cloud service brokers use service broker routing policies to select the proper datacenter.

In this paper, a new service broker routing policy is proposed for suitable datacenter selection. The proposed service broker policy tries to select the datacenter with the lowest response time, datacenter processing time, and total cost to satisfy users' needs. It extended the existing SPBR policy and was implemented on the CloudAnalyst simulation framework. The proposed service broker routing policy performs to achieve a suitable datacenter based on cost and response time. The cost is considered based on both the virtual machine (VM) and the data transfer cost. And the response time is considered based on the number of virtual machines in each datacenter. In this way, the proposed service broker routing policy is able to choose the most suitable datacenter with better processing time and response time. Again, the objective of this system is described as follows:

- To handle the issue of random selection of datacenter
- To reduce response time of user base
- To reduce datacenter processing time

The rest of the documents will be described as follows. Section 2 discusses the related works for the proposed policy. And the next section 3, describes the background theory of the proposed policy. In section 4, the proposed system is presented, and the simulation and performance results of the proposed policy are discussed in section 5. Finally, the presentation of the system is concluded in section 6.

2. LITERATURE REVIEWS

In order to satisfy users' requirements and use computing resources effectively, researchers proposed various types of service broker policies in order to select the appropriate datacenter and to enhance cloud performance.

Some proposed service broker routing policies contemplate the total cost of the datacenter as a factor in the decision [6, 7, 12, 23]; some aim to minimize the response time [2, 8, 9, 20]; and others focus on the execution time [18, 24]. Additionally, Manasrah et al. proposed an optimized service broker policy using a differential evolution algorithm [13] and a variable service broker routing policy (VSBRP) [14] to achieve an acceptable response time, execution time, and cost. Moreover, D. Arya and M. Dave also proposed a priority-based service broker routing policy [3] to select the proper datacenter by considering these three performance parameters. Therefore, depending on the response time, datacenter processing time, and total cost of the datacenter in use, the standard service broker routing policies determine the suitable datacenter.

A new cloud service broker routing based on fuzzy [1] was proposed by Al-Tarawneh and Al-Mousa in 2019 to focus on cost and performance in order to be a user priority and to make datacenter selection. However, it could not achieve a good balance between cost and performance. Also, Benlalia et al. (2019) [4] proposed a service broker policy to select a suitable datacenter by considering virtual machine cost and threshold value as the main parameters. However, the idea of this policy is to set the threshold value manually, which has a negative impact on performance. Similarly, Kofahi et al. (2019) [11] proposed a service broker policy using vector space model and multi-objective optimization technique to enhance the datacenter selection process. In this policy, it requests users for the requirements they want and then selects the matching datacenter for them. In real life, it is rare for users to want to know the specifications of datacenters.

Rafieyan et al. (2020) [22] fixed issues with datacenter randomization of service proximity-based routing policy based on distances obtained by using KNN. Since this policy did not consider the state of the datacenter, the selected datacenter may have the highest job response time. Also, Khan (2020) [10] proposed a normalization-based hybrid service broker (NHSB) policy. This policy considered virtual machine specifications, datacenter specifications, and cost as parameters. The obtained value that is achieved by using these parameters is normalized and the datacenter with the lowest normalized values is selected.

Additionally, Y. Sanjalawe et al. (2021) [25] proposed a service broker policy using a modified differential evolution to improve

performance and select suitable datacenter. This policy was intended to reduce response time, datacenter processing time, and total cost, but the result was only a trivial decrease. Mishra et al. (2021) [16] proposed a service broker policy based on the shortest job first algorithm for placing cloudlets on virtual machines in a datacenter for better utilization of resources. In this policy, datacenter processing time was considered, but response time and total cost were not considered well.

More recently, Radi et al. (2022) [21] proposed a user-priority service broker policy using VIKOR. This policy used dynamic factors such as cost, load, network, and user requirements to make an appropriate datacenter selection. As a result, this policy was able to decrease the response time and total cost, but the datacenter processing time was not properly considered. Additionally, J. Bisht and V. V. Subrahmanyam (2022) [5] modified two existing broker policies: service proximity-based routing policy and optimized response time policy. As a result, the two new policies they modified gave better response time and datacenter processing time than the two existing policies, but the total cost was not considered.

Therefore, the service broker routing policies mentioned above selected the appropriate datacenter based on certain criteria: response time, datacenter processing time, and total cost. This paper presents a new service broker routing policy that uses the proportion of virtual machine weights for each datacenter [19] and total cost, which is calculated by data transfer cost and virtual machine processing cost, to optimally choose a datacenter.

3. CLOUD SERVICE BROKER

Figuring out the best way to access, implement, and manage computing resources in the cloud can be a confusing issue for cloud users. To handle this issue, the role of cloud service brokers has become important. Cloud service brokers can help users get what they want and use computing resources efficiently. There are three primary areas including aggregation, integration, and customization brokerage in cloud service brokers. Aggregation is enabling the consumption of the cloud by end users via a cloud application marketplace approved by the company. Integration is ensuring cloud applications exchange data with each other and with on-premise applications to orchestrate business processes. Customization is augmenting cloud services with changes to data schema or enhanced security and compliance [27]. And Service brokers act as intermediaries between cloud users and service providers, and decide which datacenter should provide the requested service. In other words, service brokers are responsible for controlling critical factors between a cloud user and the datacenter using service broker policies such as:

- Service Proximity Based Routing Policy (SPBR)-This policy selects the datacenter closest to the user request based on transmission parameters such as network latency and bandwidth.
- Performance Optimized Routing-This policy monitors the performance of all datacenters and selects the datacenter with the best response time when a user request arrives. The Service Proximity Based Routing Policy is extended by this policy.
- Dynamically reconfiguring router-This policy uses the basic concept of service proximity-based routing. But this policy performs decreasing and increasing the number of virtual machines allocated to a datacenter based on the workload and selects the datacenter with the best processing time.

4. CLOUDANALYST

CloudAnalyst is a Cloud simulation tool developed on the Java platform for the simulation of large-scale cloud applications with the purpose to study and analyze the behavior of such applications under various deployment scenarios. It extends the functionality of the CloudSim toolkit through the introduction of concepts that model the Internet and Internet application behaviour. It allows the description of application workloads including information on the geographic location of users generating traffic, the location of data centres, the number of users and datacentres, and the number of resources in each datacentre. Provided with this information, metrics such as the response and processing time of requests are generated. The main features of CloudAnalyst are: the

easy-to-use Graphical User Interface, the ability to define a simulation with a high degree of configurability and flexibility, the repeatability of experiments, its graphical output, the use of consolidated technology, and ease of extension [26].

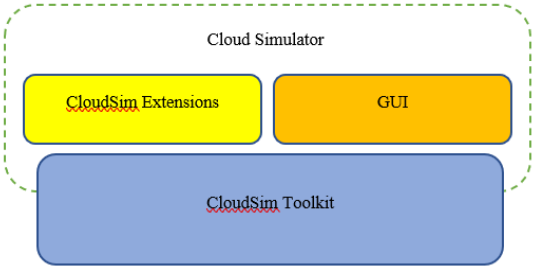


Figure 1. CloudAnalyst Architecture

5. PROPOSED SYSTEM

To be handling the issues of the existing SPBR policy, we proposed a new service broker routing policy based on the weight and the total cost of each datacenter to overcome the datacenter random selection issues. The proposed service broker routing policy selects the most suitable datacenter based mainly on the number of virtual machines in each datacenter and the total cost of each datacenter. As a result, the proposed policy focuses on the selection of the suitable datacenter with better response time, processing time and cost.

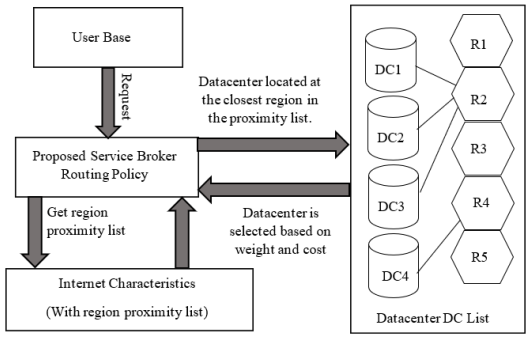


Figure 2. System Flow Diagram

In the system flow diagram, userbases are responsible for the group of users that can access the cloud datacenters through the Internet from different regions. The system does not have a fixed number of datacenters or userbases. However, the service broker is assumed to have complete information about the number of datacenters or userbases at any given time. When the userbase generates requests, the proposed policy operates to get the suitable datacenter with the lowest response time and total cost. The selection of the most suitable datacenter is performed by using the weight value and total cost of each datacenter. The policy identifies the optimal values of each datacenter by using a multi-objective optimization and calculates the average values of each datacenter. Finally, the system selects the datacenter with the lowest average value.

Response time and total cost are important factors for assessing the performance of the cloud environment. Therefore, the weight value is considered to get the best response time datacenter based on equation 1.

$$W_i = \frac{no\ of\ VM\ on\ each\ DC_i\ from\ closest\ region}{no\ of\ VM\ in\ all\ DC\ from\ closest\ region} \quad (1)$$

Where, W_i refers to the weight of each datacenter, DC refers to the datacenter, DC_i refers to each datacenter, and VM refers to the virtual machine. Then the total cost includes both the virtual machine cost and the data transfer cost and can be calculated based on equation 2.

$$C_i = \left(\frac{MIPS_{total}}{VM_{MIPS}} \right) \times VM_{cost} + UDS \times DT_{cost} \quad (2)$$

Where, C_i refers to the total cost of each datacenter that considers both data transfer and virtual machine processing cost, $\left(\frac{MIPStotal}{VM\ MIPS} \right)$ refers to the total number of instructions per average processing power assigned to the datacenter. VM_{cost} denotes the cost of each datacenter's virtual machine; UDS denotes the size of requests; and DT_{cost} denotes the data transfer cost. Then, we calculate the optimal values by using weight and total cost based on equation 3.

$$Opt_i = w1 \times W_i + w2 \times C_i \quad (3)$$

Where, Opt_i refers to the optimal value of each datacenter, W_i refers to the weight value of each datacenter, and C_i refers to the total cost value of each datacenter. According to the general formula of multi-objective scalarization, the coefficient values will determine the solution of the fitness function and show the priority. The larger the coefficient value, the higher the priority, and the sum of coefficient values must be 1. Therefore, w1 and w2 are the coefficient values. Then we calculate the total optimal value and average values of each datacenter by using equation 4 and 5, respectively.

$$totalOpt_i = \sum_{i=1}^{N} Opt_i \quad (4)$$

$$Avg_i = \frac{totalOpt_i}{N} \quad (5)$$

Where, $totalOpt_i$ refers to the sum of optimal values of each datacenter, Opt_i refers to the optimal value for each datacenter, and N is the number of intervals. Avg_i refers to the average value of each datacenter. When the average values of all datacenters have been calculated, the system selects the datacenter with the smallest average value.

Table 1. Notations Used in Proposed Service Broker Routing Policy

Notation	Description
DC	Datacenter List
UB	User Request List
R	Region List
W	Weight Value
C	Cost Value
Opt	Optimal Value
totalOpt	Total Optimal Value
Avg	Average Value

Algorithm: Proposed Service Broker Routing Policy

Input: Datacenter List DC_i = {DC_1, DC_2, …, DC_n}, User Request List UB_i = {UB_1, UB_2, …, UB_n}, Region List R_i = {R_1, R_2, …, R_6}
Output: Optimal Data Center List DC_i

Get a region proximity list
if there is new User Request **UB_i then** load Datacenter List **DC_i** of closest region
  if there is more than one Datacenter **DC_i** at that region **R_i then**
    for each Datacenter **DC_i** at that region **R_i do**
      Calculate the weight value **W_i** of each Datacenter **DC_i** using (1)
      Calculate the cost value **C_i** of each Datacenter **DC_i** using (2)
      Calculate the optimal value **Opt_i** of each Datacenter **DC_i** using (3)
      Find the total optimal value **$totalOpt_i$** of each Datacenter **DC_i** using (4)
      Calculate the average value **Avg_i** of each Datacenter **DC_i** using (5)
    end for
    Sort the average value **Avg_i** of each Datacenter **DC_i** in ascending order
    Select the smallest average value **Avg_i** and its Datacenter **DC_i**
    Return Datacenter **DC_i**
  else
    Return Datacenter **DC_i** according to SPBR Policy
  end if
end if

6. Results And Discussion

The CloudAnalyst simulation framework was used to evaluate the proposed service broker routing policy and compare its response time, datacenter processing time, and total cost with the existing SPBR policy. The network connectivity values, such as delay and bandwidth, are fixed for the experiments according to the CloudAnalyst.

6.1. Simulation Configuration

Userbase Configuration: The userbase represents a large group of users for the efficiency of the simulation, but it is considered to be the single user in the model, and the userbase generates requests in the simulation. And the following table is assumed as the userbase configuration.

Table 2. Userbase Configuration

Name	Region	Peak Hour (GTM)	Avg Peak Users
UB1	0	3-9	1000-100
UB2	1	3-9	1000-100
UB3	2	3-9	1000-100

Datacenter Configuration: The responsibility of the datacenter is to manage the data activities and to route the user requests from the userbase to the virtual machine. And, seven datacenters in the same region are considered, and each datacenter has the same hardware characteristics and only a different number of virtual machines to evaluate the proposed service broker policy. Based on virtual machine cost and data transfer cost, two datacenter configurations were assumed as in Tables 3 and 4.

Table 3. Datacenter configuration 1 with the same virtual machine cost and data transfer cost

Data-center	No. of virtual machines	Cost per virtual machines/hr	Data transfer cost/Gb
DC1	80	0.1	0.1
DC2	50		
DC3	10		
DC4	100		
DC5	70		
DC6	60		
DC7	90		

Table 4. Datacenter configuration 2 with the different virtual machine cost and data transfer cost

Data-center	No. of virtual machines	Cost per virtual machines/hr	Data transfer cost/Gb
DC1	80	0.05	0.05
DC2	50	0.1	0.1
DC3	10	0.12	0.12
DC4	100	0.15	0.15
DC5	70	0.126	0.1
DC6	60	0.126	0.1
DC7	90	0.126	0.1

Other Configuration: Table 5 lists the additional variables used in simulation.

Table 5. other configurations

Factors	Values
Simulation Duration	24 hrs
User Grouping Factor in Userbase	10
Request Grouping Factor	10
Executable instruction length per request	100
Load balancing Policy	Round Robin
Simulation Duration	60 min
Image size	10000
Memory	512
Bandwidth	1000
Datacenter Architecture	X86
Datacenter processor	4
Datacenter Operating system	Linux
Datacenter virtual machine monitor	Xen

6.2. Result and Performance Analysis

Three scenarios are used for measuring the performance. If a user request is to be handled by one or more datacenters, we can observe that the proposed service broker routing policy gives us an improvement in the overall response time (RT), datacenter processing time (PT) compared to the existing SPBR policy in Tables 6 and 7. Thus, the proposed service broker routing policy leads to more accurate datacenter selection.

By using the datacenter configuration with the same virtual machine cost and data transfer cost as in Table 3, the datacenter selection is prepared to evaluate the performance of the proposed service broker policy. In this case, two different policies have been run with the above configurations and UB1 requested the appropriate datacenter.

Table 6. Experimental Results for datacenter configuration 1

Scenario	Policy	RT	PT
UB1 must be handled by three DC	proposed	496.33	34.72
	SPBR	501.34	34.87
UB1 must be handled by five DCs	proposed	495.71	43.98
	SPBR	501.72	44.24
UB1 must be handled	proposed	496.81	46.37

by seven DCs	SPBR	501.81	46.48

According to the experimental results Table 6, one userbase requests the appropriate datacenter, and the results give that the proposed service broker routing policy is better than the existing SPBR policy. The result is evaluated

In another case, we used the datacenter configuration with the different virtual machine costs and data transfer costs in Table 4 to compare the overall response time, and datacenter processing time between the two policies. All userbases requested the appropriate datacenter. The performance parameters of each datacenter are shown in Table 7.

Table 7. Experimental Results for datacenter configuration 2

Scenario	Policy	RT	PT
All UBs must be handled by three DC	proposed	328.42	32.84
	SPBR	334.43	32.87
All UBs must be handled by five DCs	proposed	324.80	42.13
	SPBR	334.80	42.15
All UBs must be handled by seven DCs	proposed	329.89	44.27
	SPBR	334.89	44.29

In Table 7, the proposed service broker routing policy is evaluated with all userbase. In all scenarios, the overall response time of the proposed policy is better than the existing policy. But the processing time is only slightly decreased.

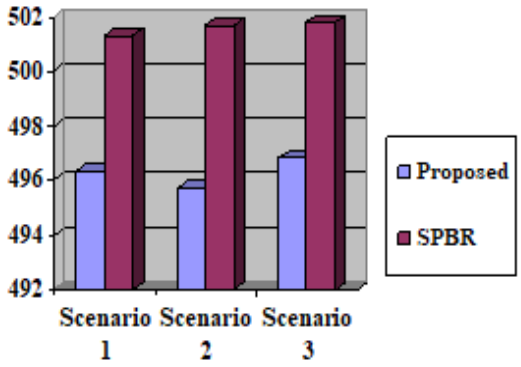


Figure 3. Overall Response Time of the Proposed Policy and Existing SPBR that are tested with Datacenter Configuration 1

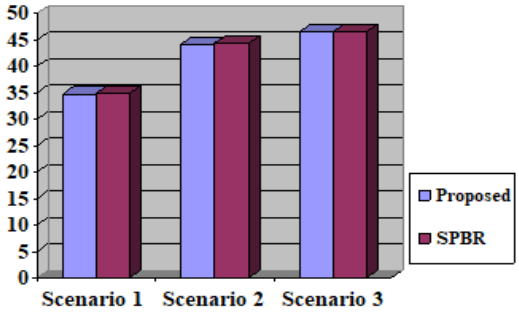


Figure 4. Processing Time of the Proposed Policy and Existing SPBR that are tested with Datacenter Configuration 1

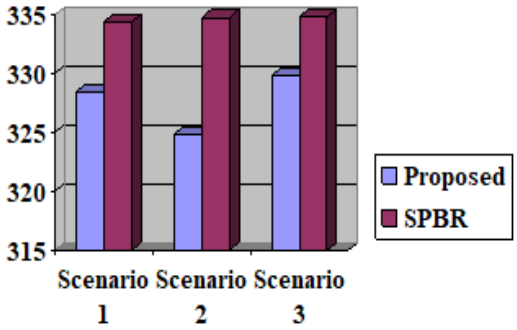


Figure 5. Overall Response Time of the Proposed Policy and Existing SPBR that are tested with Datacenter Configuration 2

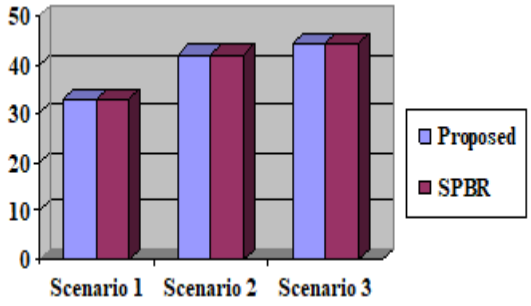


Figure 6. Processing Time of the Proposed Policy and Existing SPBR that are tested with Datacenter Configuration 2

7. CONCLUSIONS

In the existing SPBR, there were some issues due to selecting a random datacenter. The proposed Service Broker Routing Policy is proposed to minimize the datacenter processing and response time of user's requests. The proposed policy extends two behavior of the SPBR policy to select the datacenter in an efficient way. The proposed policy adds the VM weight and DC cost equations of the old policy. The proposed policy is evaluated and compared with the existing SPBR policy using the CloudAnalyst simulator. It is concluded that the proposed policy is better than the existing SPBR policy because the datacenter selection depends on the overall response time, and datacenter processing time, via the simulation results. The proposed policy can be improved in the future by the introduction of a new equation for response time, datacenter processing time, and the cost of each datacenter.

REFERENCES

[1] M. Al-Tarawneh, A. Al-mousa, "Adaptive User-Oriented Fuzzy-Based Service Broker for Cloud Services", *Journal of King Saud University - Computer and Information Sciences*, 2019.

[2] S. Ahmed, "Enhanced Proximity-Based Routing Policy for Service Brokering in Cloud Computing", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 2, Issue 2, pp.1453-1455, 2012.

[3] D. Arya and M. Dave, "Priority Based Service Broker Policy for Fog Computing Environment", *D. Singh et al. (Eds.): ICAICR, 2017, CCIS 712*, pp. 84–93, 2017.

[4] Z. Benlalia, A. Beni-hssane, K. Abouelmehdi and A. Ezati, "A new service broker algorithm optimizing the cost and response time for cloud computing*", International Symposium on Machine Learning and Big Data Analytics for Cybersecurity and Privacy (MLBDACP)*, Leuven, Belgium, pp. 992–997, 2019.

[5] J. Bisht and V. V. Subrahmanyam, "Improvising service broker policies in fog integrated cloud environment", *International Journal of Communication Networks and Distributed Systems*, Vol. 28, No. 5, 2022.

[6] D. Chudasama, N. Trivedi and R. Sinha, "Cost effective selection of Data center by Proximity-Based Routing Policy for Service Brokering in Cloud Environment", *International Journal of Computer Technology & Applications*, Vol 3 (6), pp. 2057-2059, 2012.

[7] A. Jaikar, G. R. Kim and S. Y. Noh, "Effective Data Center Selection Algorithm for a Federated Cloud", *Advanced Science and Technology Letters*, Vol.35, pp. 66-69, 2013.

[8] D. Kapgate, "Improved Round Robin Algorithm for Data Center Selection in Cloud Computing", *International Journal of Engineering Sciences & Research Technology*, Vol.3(2), pp. 686-691, 2014.

[9] D. Kapgate, "Weighted Moving Average Forecast Model based Prediction Service Broker Algorithm for Cloud Computing*", International Journal of Computer Science and Mobile Computing*, Vol.3 Issue.2, pp. 71-79, 2014.

[10] M. A. Khan, "Optimized hybrid service brokering for multi-cloud architectures", *The Journal of Supercomputing*, 2020.

[11] N. A. Kofahi, T. Alsmadi, M. Barhoush, and M. A. Al-Shannaq, "Priority-Based and Optimized Data Center Selection in Cloud Computing", *Arabian Journal for Science and Engineering*, pp. 9275–9290, 2019.

[12] D. Limbani and B. Oza, "A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation", *Int.J.Computer Technology & Applications*, Vol 3 (3), pp. 1082-1087, 2012.

[13] A. M. Manasrah, A. Aldomi and B. B. Gupta, "An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment", *Cluster Computing*, 2017.

[14] A. M. Manasrah, T. Smadi, A. ALmomani, "A Variable Service Broker Routing Policy for data center selection in cloud analyst", *Journal of King Saud University - Computer and Information Sciences*, 2015.

[15] P. Mell and T. Grance, "The NIST Definition of Cloud Computing".

[16] N. K. Mishra, P. Himthani and G. P. Dubey, "Priority-based Shortest Job First Broker Policy for Cloud Computing Environments",*Proceedings of International Conference on Communication and Computational Technologies*, pp. 279 – 290, 2022.

[17] R. K. Mishra and S. N. Bhukya, "Service Broker Algorithm for Cloud-Analyst", *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 5 (3), pp. 3957-3962, 2014.

[18] R. K. Naha and M. Othman, "Cost-aware service brokering and performance sentient load balancing algorithms in the cloud", *Journal of Network and Computer Applications*, 75, pp. 47–57, 2016.

[19] S. Nandwani, M. Achhra, R. Shah, A. Tamrakar, K. Joshi and S. Raksha, "Weight-Based Data Center Selection Algorithm in Cloud Computing Environment", *Artificial Intelligence and Evolutionary Computations in Engineering Systems, Advances in Intelligent Systems and Computing*, 394, pp. 515-525, 2017.

[20] M. Radi, "Weighted Round Robin Policy for Service Brokers in a Cloud Environment", *The International Arab Conference on Information Technology (ACIT2014)*, pp. 45-49, 2014.

[21] M. Radi, A. A. Alwan, A. Z. Abualkishik, A. Marks and Y. Gulzar, "Efficient and Cost-effective Service Broker Policy Based on User Priority in VIKOR for Cloud Computing", *The Scientific Journal of King Faisal University: Basic and Applied Sciences*, Vol.23, Issues 1, 2022.

[22] E. Rafieyan, R. Khorsand and M. Ramezanpour, "An Adaptive Scheduling Approach based on Integrated Best-worst and VIKOR for Cloud computing", *Computers & Industrial Engineering*, 2020.

[23] R. Rathi, V. Sharma and S. K. Bola, "Round-Robin Data Center Selection in Single Region for Service Proximity Service Broker in CloudAnalyst", *International Journal of Computers & Technology*, Vol. 4, No. 2, pp. 254-260, 2013.

[24] P. M. Rekha and M. Dakshayini, "Service Broker Routing Policies in Cloud Environment: A Survey", *International Journal of Advances in Engineering & Technology*, Vol. 6, Issue 6, pp. 2717-2723, 2014.

[25] Y. Sanjalawe, M. Anbar, S. Al-E'mari, R. Abdullah, I. Hasbullah and M. Aladaileh, "Cloud Data Center Selection Using a Modified Differential Evolution", *Computers,Materials & Continua*, Vol.69, No.3, pp. 3179 – 3204, 2021.

[26] B. Wickremasinghe, "CloudAnalyst: a CloudSim-based tool for modeling and analysis of large-scale cloud computing environment", *MEDC Project Report, University of Melbourne (2009).*

[27] https://www.ques10.com/p/30841/explain-cloud-services-brokeragecsb/

[28] https://www.techtarget.com/searchcloudcomputing/definition/cloud-computing

[29] https://unctad.org/system/files/official-document/ier2013_en.pdf

Authors

Biography

First A. U Kaung Myat Soe was graduaded B.C.Sc(Q) in 2020 and M.C.Sc in 2023. Now, he works at the Faculty of Computer Science in University of Computer Studies (Mandalay). He is interested in Cloud Computing and Artificial Intelligence (AI).

LEXICON BASED PUBLIC EMOTION AND SENTIMENT ANALYSIS ON COVID-19 VACCINATION

Ngun Zar Tial[1], and Thiri Kyaw[2]

[1,2] Faculty of Computer Science, University of Computer Studies (Monywa)
ngunzarti@ucsmonywa.edu.mm, thirikyaw@ucsmonywa.edu.mm

ABSTRACT

The recent Coronavirus Infectious Disease 2019 (COVID-19) pandemic has caused much impact on people lives. The enormous increase in the number of user-generated contents on social media has become an essential tool to extract public's sentiment and emotion. The system aims to find out the public sentiment and emotion on Covid-19 vaccination through user generated text based on tweets. The system uses Valence Aware Dictionary and sEntinment Reasoning (VADER) lexicon for tweet annotation and divides feature extraction into two parts like sentiment word extraction and emotion word extraction. The system applies both VADER and TextBlob lexicons words as a sentiment word for sentiment feature extraction. National Research Council (NRC) Word-Emotion Association Lexicon is applied to emotion feature extraction because this lexicon has eight emotion such as anger, fear, joy, disgust, sadness, anticipation, trust and surprise. The system combines emotion feature with sentiment feature to improve the accuracy of the classifier models. Multinomial Naïve Bayes (MNB), Logistic Regression (LG) and Support Vector Machine (SVM) models are used to compare the accuracy of the system. Support Vector Machine get the highest accuracy, 92.36%.

KEYWORDS

Sentiment and emotion analysis; Sentiment feature extraction; Emotion feature extraction; Logistic Regression; Multinomial Naïve Bayes; Support Vector Machine

1. INTRODUCTION

The outbreak Coronavirus started in December 2019 and reached other countries from China which is the first occurring place. This disease has threatened people's lives in all time because the virus can spread from one to another easily. The government has declared many strict rules to fight the pandemic such as social distancing, wearing masks, and washing hand gel after touching something, closing of public places like beaches, restaurants, and schools etc. But this protective rules are not well affective to reduce the spreading of virus and every country in the world is battling to control the pandemic. During this pandemic, people badly suffer mental, social and healthcare problems. As a result, some people quit or lose from their works and many student couldn't attend their class time. Moreover, the price of goods simultaneously increases and some medicine and food also shortage because some people abnormally bought food and medicine. This is an interesting fact to analyse the public opinion and sentiment analysis.

There is nothing important except to get a Covid-19 vaccine as soon as possible. Although the governments are taking different measures in response during crisis, the best way to encourage citizens are difficult. Demographic, social and behavioural factors make vaccine hesitancy. According to WHO reports, there are seven covid-19 vaccines for emergency use that are Moderna, AstraZeneca or Oxford, Sputnik V, Sinopharm, Sinovac and Johnson & Johnson.

During this crisis, people's life style becomes changing, stayed at home and turned towards social media to educate and

communicate their feelings or thoughts. In early 2020, social media became one of the most reliable sources for the latest coronavirus information and important tool for conversation chats or interacting with friends and family because people was associated with widespread lockdowns and social distancing and they need to express their mental situation. There are many emotion or sentiment about Covid-19 vaccination. Some people are afraid of vaccine uptake whether the vaccine is safety or reliable. Some misinformation can make public not to vaccinate. Having varied Covid-19 vaccine can increase many anxieties because different people have different feelings depending on the situation. Knowing public's opinion on Covid vaccine is important to counteract the virus.

Social media such as Twitter, Facebook allow people to post their opinion and feeling freely. Today, many people use social media to share their opinion or emotion that will be negative or positive feeling. So social media have become the most useful framework for research because there are different age users. The system detects public sentiment and emotion on Covid-19 vaccination who use Twitter. Sentiment analysis from social media data is one of the highest emerging research fields. The sentiment analysis can be done by using different approaches that is lexicon-based approach, machine learning or deep learning approach and hybrid approach. This system is implemented during Covid vaccination period using lexicon-based approach and machine learning approach (hybrid approach) that is the dataset used in this system is annotated using VADER lexicon and machine learning algorithms are used for system performance.

Words or phrases which represent positive, neutral and negative play a vital role in sentiment analysis tasks. VADER and TextBlob lexicons are applied to sentiment analysis. Besides, NRC lexicon is an emotion lexicon with eight different emotion types and used for emotion feature extraction. This system builds a useful framework for sentiment and emotion on Covid-19 vaccination dataset using classification models. The system compares three classification models which are Multinomial Naïve Bayes, Logistic Regression and Support Vector Machine.

2. RELATED WORK

The feeling of people has been assessed through several studies before and after the development of vaccinations. The authors in [7] analyzed the public sentiment toward vaccination based on the data of all vaccines including Pfizer, Moderna, Oxford/ AstraZeneca, Covaxin, Sputnik V, Sinopharm and Sinovac to understand the public opinion. They collected the dataset from Kaggle and classify the tweet text into positive, negative and neutral by using VADER. They further analyzed the timeline of the tweets to visible the fluctuation of emotions or sentiments over time during the pandemic. To assess the performance of models, they utilized deep learning such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM). They get 90.59% accuracy in long short-term memory (LSTM) and 90.83% accuracy in bidirectional LSTM (Bi-LSTM) model with training 75% and testing 25% splitting. Bi-LSTM gives a slightly better performance than LSTM in the sentiment analysis task.

Authors in [2] proposed a methodology that combines the emotion lexicon with the classification model to enhance the accuracy of the models in sentiment analysis. They used Text-CNN, Bi-GRU and BERT for classification and combined emotion lexicon in system feature vectors. They found that VnEmoLex lexicon has improved the performance of classification model on three datasets. Besides, they showed that text preprocessing is an essential role in boosting the classifier model.

Vaccination campaigns are the essential process to reduce the covid-19 infection and fatality risks. Authors in [1] proposed a methodology to analyze the global perceptions and perspectives towards

Covid-19 vaccination using Twitter dataset. For performing various natural language processing on raw text, Lexicon-based method was applied. VADER, TextBlob and AFINN lexicon were used to annotate tweets. This study relied on two techniques: lexicon based methods for annotating the dataset and machine learning and deep learning models for sentiment analysis. TextBlob labelling dataset achieved good performance in machine learning models.

Authors in [8] focused on the sentiment analysis regarding covid-19 vaccine on Twitter. They used VADER and TextBlob for sentiment analysis and Latent Dirichlet Allocation (LDA) to determine the popular themes. They collected tweets from November 1 to December 16 in 2020 by using 14 keywords via Twitter API. They get seven topics in identifying popular themes and salient topics with 12 iterations in LDA. Positive sentiment tweets were the highest in both lexicon and TextBlob get more negative sentiment. Among seven topics, Healthcare environment was the maximum number of tweets and Covid and US elections was the least dominance.

Authors in [9] used MNB, RF, SVM, Bi-LSTM, CNN and BERT on a Covid-19 vaccination dataset from November 9, 2020 to December 8, 2020. They used stance detection approach that is evaluated as a favor, against and neutral. For performance measure, precision, recall, accuracy and f-measure was used and evaluated through 5-fold cross validation procedure. BERT model gets the highest accuracy (78.94%) and most of the tweets was neutral stance.
To understand the mental health of the people and take necessary measures against covid-19, the authors in [3] used NRC Word-Emotion Association Lexicon (EmoLex). The dataset used in this system was related to covid-19 between January 22 and April 15, 2020. They classified emotion with joy, sadness, anger, fear, disgust and surprise. They get trust emotion highest and fear emotion second highest. The number of positive and negative sentiment tweets were almost equal.

Authors in [11] performed sentiment analysis in India during lockdown. They used TextBlob and VADER for annotating tweets and count vectorizer for feature extraction. Eight different classifiers are applied on a dataset from April 5 to April 15, 2020. They used k-fold cross validation procedure with k=10 for each unigrams, bigrams and trigrams and achieved 84.4% with LinerSVM unigram. Positive sentiment was the highest.

3. METHODS AND MATERIALS

Lexicon is the vocabulary of language or the catalog of words. Lexicon based method uses sentiment dictionary with opinion words and match them with data to determine polarity. They assign sentiment scores to the opinion words by describing how positive, negative and objective the words included in the dictionary are. Lexicon based approach calculates the orientation of words or phrases. It counts the number of positive and negative words in the given text and returns a positive sentiment if the number of positives is more than the negatives otherwise it returns a negative sentiment. For lexicon-based approaches, a sentiment is defined by its semantic orientation and the intensity of each word in the sentence. Lexicon based approach can be divided into two classes: Dictionary based approach and Corpus based approach.

3.1. Lexicons

The system uses three lexicons for analysis task. VADER is a rule-based lexicon that performs well in sentiment analysis of social media text. VADER labelled words according to their semantic orientation as positive or negative or neutral. VADER lexicon assigns polarity scores to text, such as positive, neutral, negative and compound which offers the overall sentiment. VADER's lexicon dictionary contains around 7,500 sentiment words in total. The compound score is used as a threshold value for the analysis of the text data which is the sum of all lexicon ratings normalized between -1 (most extreme negative) and 1

(most extreme positive). Negation handling is essential in sentiment analysis. The system calculates negation term in sentence by multiplying sentiment score to negation scalar (-0.74) [4].

TextBlob is a lexicon based sentiment analyzer and has some predefined rules. TextBlob is used for sentiment analysis in this system. It gives both polarity score and subjectivity score. Polarity score lies between (-1 to 1) where -1 identifies the most negative and 1 identifies the most positive words. Subjectivity score lies between (0 and 1) that refers to how objective or subjective a text is. It employs averaging technique in calculating sentiment for a single word. For negation, it multiply sentiment score to -0.5 [10].

NRC Word-Emotion Association lexicon also called EmoLex (Mohammad and Turney, 2013), uses the Plutchik (1980) 8 basic emotions. It provides over 100 languages and contains 14182 words such as 1247 (anger), 839 (anticipation), 1058 (disgust), 1476 (fear), 689 (joy), 191 (sadness), 534 (surprise), 1231 (trust), 9719 (no emotion), 2312 (positive) and 3324 (negative) [12].

3.2. Classifiers

A classification model predicts the class label from a given example of input data. It requires a training dataset and calculates data input to define class label. The system use three classification models.

Multinomial Naïve Bayes classifier is a supervised machine learning algorithm. It based on the Bayes theorem and naive independence assumptions between features. The NB classifier assumes that the features are conditionally independent of one another provided class [6]. Bayes Theorem,

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (1)$$

where, $P(c \mid x)$ = the posterior probability of the class (target) given predictor (attribute)

$P(c)$ = prior probability of class c

$P(x \mid c)$ = occurrence of predictor x given class c probability

$P(x)$ = the prior probability of predictor

$$P(\text{c}) = \frac{Nc}{N} \quad (2)$$

where, Nc = number of class in all documents

N = number of all documents

In mathematical, Logistic Regression, supervised machine learning algorithm, is used to model the probabilities of a certain class or events. Multinomial logistic regression uses the logistic regression techniques to predict the target class (more than 2 target classes) [5].

$$Logit\ (Y) = \ln(\frac{\pi}{1-\pi}) \quad (3)$$

where, π = the probability of occurrence for the outcome Y

$\frac{\pi}{1-\pi}$ = the odds of success

Support Vector Machine (SVM) is one of a machine learning algorithms usually used for classification and regression. SVM's aim is to divide hyperplane in the search space that can best separate the group of instances in one class from another class and finds the best hyperplane based on the concept of decision hyperplanes. It determines the decision boundaries in their input space or high dimensional feature space [13]. To maximize the margins, the decision hyper plane is

$$f(x) = q.\ x + p \quad (4)$$

where, q = weights vector

x = input feature vector

p = bias

There are various kernel functions such as linear, radial basis functions and sigmoid. The system uses linear function. For multiclass classification, One-Against-One Support Vector Machine is used. It constructs $M \times (M-1) / 2$ binary classifiers. It finds the subsequent class by selecting the class voted by the majority of the classifiers.

3.3. Bag of Words Model

Bag of words is a natural language processing technique and represents the text by the occurrence of words count. Algorithm 1 checks the sentiment word in

each sentence. If a word is in lexicon, the system sets the sentiment vector to one.

Algorithm 1. Feature extraction for sentiment word

```
set sentiment_vector to zero
for each sentence in the sentences
    if word is in lexicon
        set sentiment_vector to one
    endif
endfor
```

Algorithm 2 finds the emotion word in NRC lexicon. If a sentence has negation word, the system converts these emotion word to antonym and set the corresponding emotion feature vector to one.

Algorithm 2. Feature extraction for emotion word

```
set emotion_vector to zero
for each sentence in the sentences
    if sentence has negatin word
        set emotion word to antonym
    endif
    if word is in NRC lexicon
        set emotion_vector to one
    endif
endfor
```

4. PROPOSED SYSTEM

The following diagram is the overview of the proposed system for sentiment analysis on Covid vaccination. The system needs to take Covid vaccination dataset as input. These tweets include lot of noise and uninformative parts such as http link, @name etc. The system discards noise data during data preprocessing stage to be better performance for the next stage. Cleaned tweets are labelled by using VADER lexicon to categorize the text into positive, negative and neutral. After tweet annotation, the system extracts feature for both sentiment and emotion. Then the system combines these two features to employ in three classification models. Finally, the system compares three classification models with performance results and which classification model get high accuracy with which feature extraction.

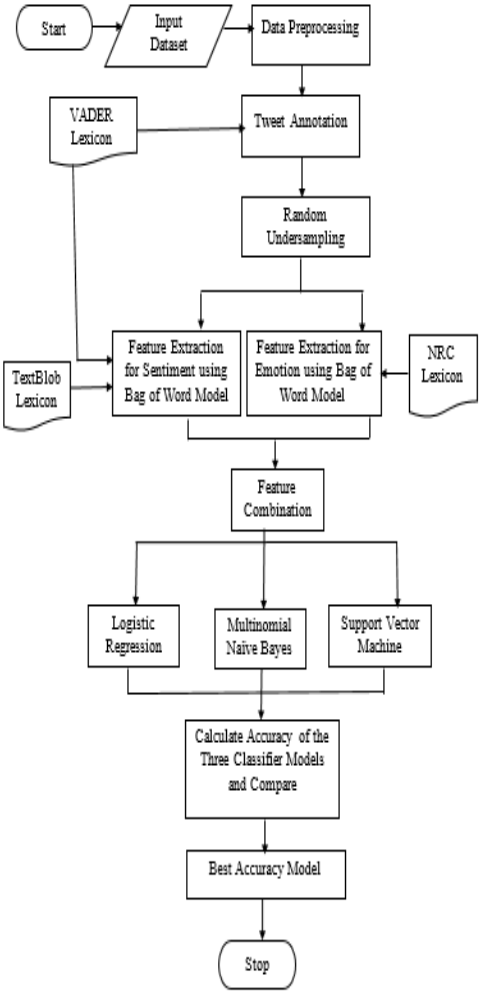


Figure 1. System Flow Diagram for Emotion and Sentiment Analysis on Covid Vaccination

4.1. Dataset Description

In this system, the dataset is collected from Kaggle which contains various types of tweets related to Covid vaccines. In dataset, it includes from December 20, 2020 to September 7, 2021 with 189,055 original data. The dataset contains 16 attributes and is shown in table 1. The system uses only text attribute.

Table 1. Description of the Dataset

id	Identification number of the user
user_name	Name of the user
user_location	Location of the user
user_description	Description of the user
user_created	User created time
user_followers	Number of user's followers
user_friends	Number of user's friends
user_favourites	Number of user's favorites
user_verified	Account verification
date	User's tweeted date
text	User's tweeted text
hashtags	Word or keyword phrase that user interested in
source	Tweet's source how the tweet posted
retweets	Reposting of a tweet
favorites	Indication how many times this tweet has been liked by Twitter users
is_retweet	Representation of the original tweet that was retweeted

4.2. Data Preprocessing

Data preprocessing is one of the component of data preparation and process on raw data in order to prepare for another data processing procedure. It is an essential preliminary step for the data mining process to remove missing value, noisy data and other inconsistencies. The steps of data preprocessing are: (1) convert lowercase and add contractions, (2) tokenize, remove emoji, @name and link, (3) remove special character and stop-words and (4) remove duplicate records.

4.2.1. Convert lowercase and add contractions

Tweet text has a different capitalization such as the beginning of sentence or their emphasized word. The common approach for simplicity in NLP tasks or text mining is to convert the text into lowercase. Converting all tweet texts into lowercasing make the process to consistent. Text has a shortened or combined two words. Negation plays an essential role in performing sentiment analysis of tweet. Thus, words like "I've", "can't" and "don't" transform into "I have", "can not", "do not", respectively.

4.2.2. Tokenize, remove emoji, @name and link

After converting lowercase, these texts are split into token for next preprocessing step. Emoji, @username, and URL links in the dataset are discarded because they do not carry much information in this system. Numbers are also removed from tweets except 19 because the research is for Coivd-19 vaccination. Moreover, number mix with word such as "1st", "2nd" etc is not eliminated because there have 1st doses, 2nd doses in Covid-19 vaccination.

4.2.3. Remove special character and stop-words

Special character such as hashtags, dollar signs etc are removed from tweet. Stop-words are the frequently appeared word in dataset such as 'the', 'a' and 'is'. These words are not important in sentiment analysis and are removed from texts in many tasks. By discarding these words, the system can only focus on the important words and reduces the number of features in feature extraction. There are 179 stop-words in NLTK (Natural Language Toolkit) ython. But this system uses 114 stop-words.

4.2.4. Remove duplicate records

The final step in data preprocessing removes duplicate records. These redundant

record does not provide information and not necessary in analysis. Therefore, these records are discarded from the dataset to simplify the classifier models.

4.3. Tweet Annotation

The system uses VADER lexicon for tweet annotation and sentiment analysis. The system assigns the polarity of positive, negative and neutral by using this lexicon. The system removes 47737 tweets after data preprocessing stage and result is shown in table 2. After tweet annotation, the sentiment classification result is shown in table 3. Neutral sentiment is the highest and negative sentiment is the lowest.

Table 2. Number of Tweets

Original dataset	189,055
After preprocessing	141,318

Table 3. Sentiment Classification of Covid Vaccination Tweets

Lexicon	**Class Label**		
	Positive	**Negative**	**Neutral**
VADER	52097	26142	63079

4.4. Random Undersampling

Undersampling and oversampling are techniques used to control the problem of unbalanced classes in a dataset. This technique is needed in order to avoid overfitting of data with a majority class at the expense of other classes whether it's one or multiple. Random undersampling delete events randomly from the majority class. This method is appropriate if there is plenty of data for an accurate analysis. After random undersampling, the system uses 78426 tweets for sentiment and emotion analysis.

4.5. Feature Extraction

Before applying any machine learning classification algorithm, tweet needs to be transformed into a numerical feature vector which means that feature extraction is a method of discovering significant characteristics of data. The simplest method to do this is to apply the vector space representation model which means that by capturing or weighting each word based on its importance in text.

The system uses bag of word model as a feature vector. In Feature Extraction for sentiment word, the system only focuses on VADER and TextBlob lexicon dictionary words. There are 2860 dictionary words in TextBlob lexicon and 7506 dictionary words in VADER lexicon. The system discards non alphanumeric words from feature extraction and mixed word from these two lexicons. So, the system obtains 9269 dictionary words for sentiment feature extraction. But the dataset applied in this system does not need those amount of feature words. Finally, the system uses only 3505 feature words for sentiment word feature extraction process that are 2684 words from VADER and 821 words from TextBlob.

For emotion feature extraction, the system uses eight emotion such as trust, anticipation, sadness, anger, fear, joy, disgust and surprise as a feature. The system used only English language for emotion feature extraction. The result of feature extraction for sentiment and emotion is shown in table 4.

Table 4. Number of Sentiment and Emotion Feature

Number of Sentiment Features	**Number of Emotion Features**	**Number of Sentiment and Emotion Features**
3505	8	3513

4.6. Experimental Results

From table 5, Support Vector Machine outperforms than other Multinomial Naïve Bayes and Logistic Regression. The dataset used in this system is the same to the

dataset in reference [7]. They analyzed the sentiment of people by using tweet amount of 125906. Similarly, this system focus on 78426 tweets for emotion and sentiment analysis by applying NRC, VADER and TextBlob lexicons. When comparing only sentiment, the authors in [7] get the highest accuracy with Bi-LSTM 90.83% and this system get the highest accuracy with SVM 92.09%. After combining sentiment and emotion feature, SVM get the accuracy of 92.36%. Therefore, this system slightly improves the accuracy than the authors in [7]'s implemented system.

Table 5. Accuracy Result for 75-25 Splitting

Classifiers	Features	Accuracy (%)
Multinomial Naïve Bayes	Sentiment Feature	71.67
	Emotion Feature	42.03
	Combined Sentiment and Emotion Feature	74.13
Logistic Regression	Sentiment Feature	90.50
	Emotion Feature	54.64
	Combined Sentiment and Emotion Feature	90.61
Support Vector Machine	Sentiment Feature	92.09
	Emotion Feature	54.55
	Combined Sentiment and Emotion Feature	92.36

In Multinomial Naïve Bayes Model, the accuracy of combined sentiment and emotion feature gets higher accuracy than only sentiment feature. The accuracy of combined sentiment and emotion feature performs better than sentiment only feature.

5. Conclusions

The system analyzes and classify the tweets regarding Covid-19 vaccination. The proposed framework classifies tweets into three main class: positive, negative and neutral. The best performing classifier model was identified with their accuracy rates. Boosting the accuracy of the classifier model is an important role in making analysis tasks. The experimental result shows that the use of emotion word in sentiment analysis effect the performance of the classifier models. Emotion and sentiment analysis can be very useful because almost Nations are vaccinating their populations with booster doses and making new protective measures. This would be helpful for public healthcare organizations to be effective in management strategies and announce the public with reliable news by minimizing the distribution of fake news. The system only analyze on English tweet to understand the public sentiment and emotion. Moreover, the system has the ambiguity in the text such as abbreviation make the classifier model confused in detecting the actual emotion from text.

Acknowledgements

I especially thank to my supervisor, Dr Thiri Kyaw, Professor, University of Computer Studies (Monywa).

References

[1] A.A. Reshi et al., (2022) "COVID-19 Vaccination-Related Sentiments Analysis: A Case Study Using Worldwide Twitter Dataset", *MDPI*.

[2] A.L. Doan and S.T. Luu, (2022) "Improving Sentiment Analysis by Emotion Lexicon Approach on Vietnamese Texts", 4 December.

[3] A. Mathur, P. Kubde and S. Vaidya (2020) "Emotional Analysis using Twitter Data during Pandemic Situation: COVID-19", *Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES)*.

[4] C. Hutto and E. Gilbert (2014) "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text",*Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, https://ojs.aaai.org/index.php/ICWSM/article/download/14550/14399/18068

[5] C.J. Peng et al., (2002) "An Introduction to Logistic Regression Analysis and Reporting", *Journal of Educational Research*.

[6] D. Jurafsky and J.H. Martin, (2020) *Speech and Language Processing*, 30 December, pp 55-61.

[7] K.N. Alam et al., (2021) "Deep Learning-Based Sentiment Analysis of COVID- 19 Vaccination Responses from Twitter Data", *Computational and Mathematical Methods in Medicine*.

[8] K. Rahul et al., (2021) "Analysing Public Sentiments Regarding COVID-19 Vaccine on Twitter", *7th International Conference on Advanced Computing and Communication Systems (ICACCS)*.

[9] L. Coftas et al., (2021) "The Longest Month: Analyzing COVID-19 Vaccination Opinions Dynamics From Tweets in the Month Following the First Vaccine Announcement".

[10] L. Steven,(2020)"textblob Documentation", 26 April.

[11] P. Gupta et al., (2021) "Sentiment Analysis of Lockdown in India during COVID-19: A Case Study on Twitter", *IEEE transactions on computational social systems*, vol. 8, no. 4, August.

[12] S.M. Mohamma and P.D. Turney, (2013) "Crowdsourcing a Word–Emotion Association Lexicon", 28 August, https://arxiv.org/pdf/1308.6297.pdf

[13] T. Yu et.al., (2019) "Myanmar News Sentiment Analyzer using Support Vector Machine Algorithm", *International Journal of Advanced Trends in Computer Science and Engineering*, vol.8, November-December.

Authors

Biography

Ms. Ngun Zar Tial received B.C.Sc degree from University of Computer Studies, Monywa, in 2020. She is a master candidate of University of Computer Studies (Monywa). Her research interest is Natural Language Processing.

Dr. Thiri Kyaw received Ph.D(IT) from University of Computer Studies, Mandalay, in 2021. She is a professor of University of Computer Studies (Monywa). Her research interest is web mining. She has been supervising Master thesis.

SHORTEST PATH FINDING SYSTEM USING A * SEARCH ALGORITHM

Su Su Win [1]

[1] Faculty of Computer Science, University of Computer Studies (Mandalay)
susuwin.loikaw@gmail.com

ABSTRACT

*At the present time, as technologies are being revolutionized from time to time, the paper is based on the knowledge and experience in the field of artificial intelligence (AI). The shortest path algorithm is one of the most fundamental and important in different areas of industry, business. In human life, many people travel to everywhere about work, visit, study and economy. So they choose to arrive with minimum distances by shortest path. In this paper proposes, when the user gives any road map as input, stores in SQL database. If the user want to search the shortest path of any two cities (source and destination) in given road map, the system calculates the shortest path using A * search algorithm. This shortest path show the total minimum distances of the path passed of over all cities. This system can find shortest path of the world over, if the relation of the cities and distances between them is given by user. Therefore, it is widely used for various maps. As a result, using of shortest path finding system, it can save the time and cost. The implementation of the shortest path finding system is used as undirected graph. In this paper investigates the above algorithm and implements this algorithm by using Java programming languages and MySQL Database.*

KEYWORDS

*A * search algorithm, Shortest path, Artificial intelligence (AI)*

1. INTRODUCTION

As the technology grows rapidly, many people take a great interest in computer and then computer base methods are increasingly used to improve the traveling and agencies services. In computer science, graph theory plays an important role because it provides an easy and systematic way to find the optimal route for the traveling and agencies services. There are a number of ways to classify shortest path algorithms. One of the main reasons for the popularity of A * Algorithm is that it is one of the most important and useful algorithms available for optimal solutions to a large class of shortest path problem.

A star algorithm is especially use to find shortest path of given road map by user in this system. Computing a shortest path from node to another in an undirected graph is a very common task. Early shortest path work has been done by Dijkstra, Bellman and Ford, A star, Floyd and Warshall. Furthermore, A star algorithm is widely used in shortest path finding system. Algorithms for determining the through a shortest paths graph typically exploit the fact that a given shortest paths must contain other shortest paths within it.

A* is a road map searching algorithm that takes a ‘distance-to-goal +path-cost’ score into consideration. As it traverses the road map searching all neighbours, it follows lowest score path keeping a sorted priority queue of alternate path segments along the way. If any point being followed has a higher score than other encountered path segments, the higher score path is abandoned and a lower score sub path traversed instead. This continues until the goal is reached.

This system is implemented by using locations within Myanmar country as the vertices of an undirected Graph. In this system, the associated distances between each location are represented as weight of the edges of the graph. This paper of

shortest path finding system, user can input any road map, any division and everywhere as implement.

This paper is organized as follows. Section 1 is the introduction. In section 2 of this paper background theory is discussed. This contains Artificial intelligence, shortest path, Undirected Graph and A* algorithm. Result and Discussion, System design, Database Design and Implementation of the Shortest Path Finding System have also been described in section 2. Section 3 is the conclusion.

2. BACKGROUND THEORY

In this section, Artificial Intelligence, Shortest Path, Graph, Undirected Graph and A * search algorithm.

2.1. Artificial Intelligence (AI)

Artificial intelligence is the ability of a computer-enabled robotic system to process information and produce outcomes in a manner similar to the thought process of humans in learning, decision making and solving problems. The goal of AI systems is to develop systems capable of tacking complex problems in ways similar to human logic and reasoning. AI is the science and engineering of making intelligent machines, especially intelligent computer programs [1].

AI is an information technology-based computer system or machine that has the ability to complete tasks that usually require human intelligence and logical deduction. Artificial intelligence (AI) is transforming the way we live, work, and interacts. From our personal lifestyles through our social engagements to the way we conduct our private and corporate businesses, AI is altering our methodologies and changing the landscape of end products. From the age-old medical expert systems and intelligent search engines to intelligent Chabot and predictive models, the enthusiasm for AI practice is growing rapidly.

AI is still weak in soft skill such as creativity, innovation, and critical thinking, problem solving, socializing, leadership, empathy,collaboration, and communication. This is not to say that we should ignore the hard skills such as science, mathematics and engineering. Higher education should still train the students in the fundamentals of science and math, and at the same time provides opportunities and training for students to enhance their soft skills [6].

2.2. Shortest Path

The shortest path problem is amongst the most well studied in the computer science literature [5]. Shortest path algorithms are used to find from the source city to destination city. Shortest path algorithms operate on a graph, which is made up of vertices and edges that connect them. The problem of finding a path that the minimum distance (weight) between points (cities) on road map is called the shortest path problem. A path is called the shortest path if it has the shortest length from among all paths that begin and terminate in given vertices [3].The shortest path problem is a popular mathematics problem that asks for the most efficient trajectory possible given a set of points and distances that must all be visited. So this paper use shortest path finding systems by using shortest of the path from one place to another, it will give less time and money in land trading.

2.3. Graph

A * algorithm is based on graph theory. A Graph is a collection of Vertices (V) and Edges. Vertices also called as nodes. V is a finite set of nodes. Edges refer to the link between two vertices or nodes. E is a set of pairs of vertices. A weighted graph is a graph whose edges have some sort of value that is associated with them. The graph is directed graph, undirected graph or contains cycles [2]. The graph can be used to optimize traffic flow and plan transportation infrastructure.

2.4. Undirected Graph

Undirected graph or also sometimes referred to as an undirected network is a kind of a graph where the links, or edges,

do not possess any specific direction [2]. Undirected graphs are used in traffic flow optimization to model the flow of vehicles on road networks .The vertices of the graph represent telephone line, network line, road map, and the edges represent the connections between them. Many problems can be modelled using graphs with weights assigned to their edges. As use the algorithm implies new bound on both the all-pairs and single-source shortest path problems. And also it can loop the whole graph satisfy for route finding task. Therefore, undirected graph is especially used for shortest path algorithm.

2.5. A * Algorithm

A* is a generic search algorithm that can be used to find solutions for many problems, pathfinding just being one of them [4]. A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals. A* Algorithm is a simple algorithm to find the nearest route. It avoids expanding paths that are already expensive. A * search maintains the set open of so-called open nodes that have been generated but not yet expanded. This method always selects a node from open with minimum estimated cost, one of those it considers "best". This node is expanded and cost of some node n with an evaluation function of the form f (n) =g (n) +h (n).

So g (n) gives the path cost from the start node to node n. h (n) is the estimated cost of the cheapest path from n to the goal, f (n), estimated cost of the cheapest solution through n. The objective is to find the shortest path between the start node and goal node and the minimum number of squares we need to transverse in order to reach goal form. The A * algorithm to enable to find an optimal path from its starting position to goal location specified by a human operator for a given environment.

Process of A* algorithm is shown in the following.

1) Add the starting node to the open list.
2) Repeat the following:
a) Look for the lowest F cost square on the open list. This is referred as the current node.
b) Switch it to the closed list.
c) For each of the adjacent nodes to this current node. If it is not workable or if it is on the closed list, ignore it. Otherwise do the following. If it isn't on the open list, add it to the open list. Make the current square the parent of this square. Record the F, G, and H costs of the square. If it is on the open list already, check to see if this path to that node is better, using G cost as the measure. A lower G cost means that this is a better path. If so, change the parent to the current node, and recalculate the G and F scores of the node. Open list is sorted by F score.
d) Calculation is stopped: Add the target node to the closed list
3) Save the path.

2.6. Result and Discussion

In this research, the shortest path can easily be found for travelling and transportation. A * search algorithm run to find the shortest path and apply in real world. Now we like to express how we get the shortest path between the most famous two cities in Myanmar, Taunggyi as a source and Yangon as a destination.

The identification of implemented figures for graphical representation, such as Straight Line Distances (SLD) to destination (Yangon) is shown in table 1.

Table 1. Straight Line Distances

No	FROM	TO	Distance(miles)
1	Taunggyi	Yangon	399
2	Loikaw	Yangon	304
3	Pinlaung	Yangon	310
4	Kalaw	Yangon	358
5	Meiktila	Yangon	310
6	Pyawbwe	Yangon	297.6
7	Mandalay	Yangon	389
8	Naypyidaw	Yangon	228.1
9	Taungoo	Yangon	172.3
10	Phyu	Yangon	180.1
11	Bago	Yangon	44.0
12	Yangon	Yangon	0

Source=Taunggyi

Destination=Yangon

-f(Kalaw)=44.1+358=402

f(Loikaw)=92.2+304=396

-f(Taunggyi)=88.2+399=487.2

f(Pinlaung)=92.6+310=402.9

f(Pyawbwe)=105.9+297.6=403.5

f(Meiktila)=116.5+310 =426.5

-f(Kalaw)=141.7+358=499.7

f(Loikaw)=148.7+304 =452.7

-f(Kalaw)=167.7+358=525.7

f(Naypyidaw)=182.2+228.1=410.3

-f(Meiktila)=271+310 =581

f(Pyawbwe)=258.2+297.6=555.8

f(Taunggo)=253.7+172.3=426

-f(Naypyidaw)=325.2+228.1=553.3

f(Phyu)=298.2+180.1=478.3

-f(Bagoo)=378.1+44=422

-f(Yangon)=422+0=422

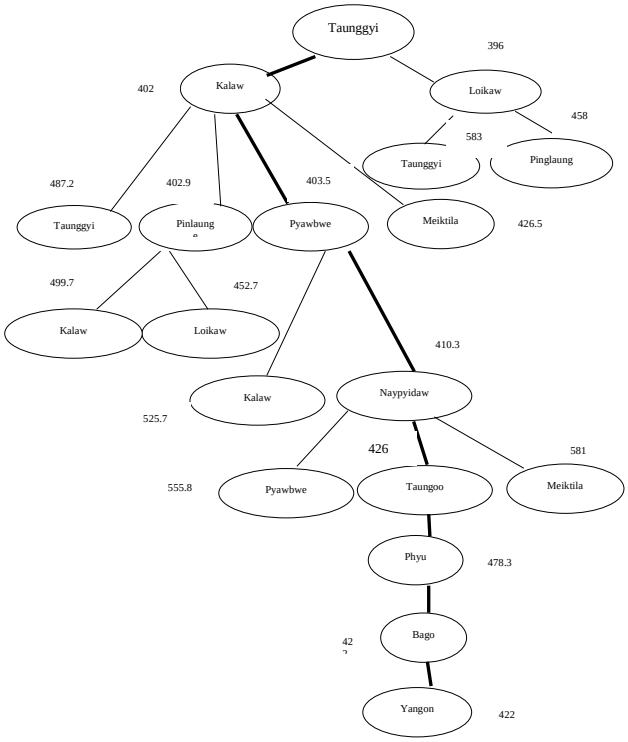


Figure 1. Graph representation of shortest path evaluated result from Taunggyi to Yangon

The above is the samples of finding shortest path from Taunggyi to Yangon these cities are exist in Myanmar. The traversed figure of finding shortest path from Taunggyi to Yangon show is figure 1. The shortest distance is 422 miles in the table with the shortest path from Taunggyi to Yangon. As a result, we can prove the implemented shortest path algorithm. A * search algorithm is a good one to be optimized in shortest path problem. We can also say that the space complexity of this algorithm is O (E), E is the number of edges because of traversing 7 edges in this evaluation as shown in figure 1.The possible paths are shown in table 2 .

Table 2. Possible Path

Source	Desitnation	Route	Distance(mile)
Taunggyi	Yangon	Taunggyi-Loikaw-Pinlaung-Pyawbwe-Naypyidaw-Taungoo-Phyu-Bago-Yangon	574.5mile
		Taunggyi-Loikaw-Pinglaung-Kalaw-Meiktila-Naypyidaw-Taungoo-Phyu-Bago-Yangon	525.5mile
		Taunggyi-Kalaw-Meiktila-Naypyidaw-Taungoo-Pyay-Bago-Yangon	445.2 mile
		Taunggyi-Kalaw-Pyawbwe-Naypyidaw-Taungoo-Phyu-Bago-Yangon	**422 mile**

2.7. System Design

This system A * algorithm is used to find the shortest path between source and destinations. Process flow of the system is shown below in Figure 2: User has to gives any road map for input by the system. And then, user has to input source city and destination city that is necessary for their distance. After input the road map, the shortest path system is processed and stored it in SQL database. It next information of new nodes can edit, insert, delete, and update their distance by this system. The information given by user has to be computed with A * algorithm that is generated by the system. Performance of the system can be search by any road map and any other division. Moreover, new road map can also as input again and again. There can plot one or more new city between existing points. Another way, user can create the desire map.

Additionally, remove out the road map entry from database. So, user has to use

above the pleasure for their travelling and trading. This is useful for reduce cost and time system consumption by using this arrangement. As a result, this will display the optimal route and solution of shortest path passed.

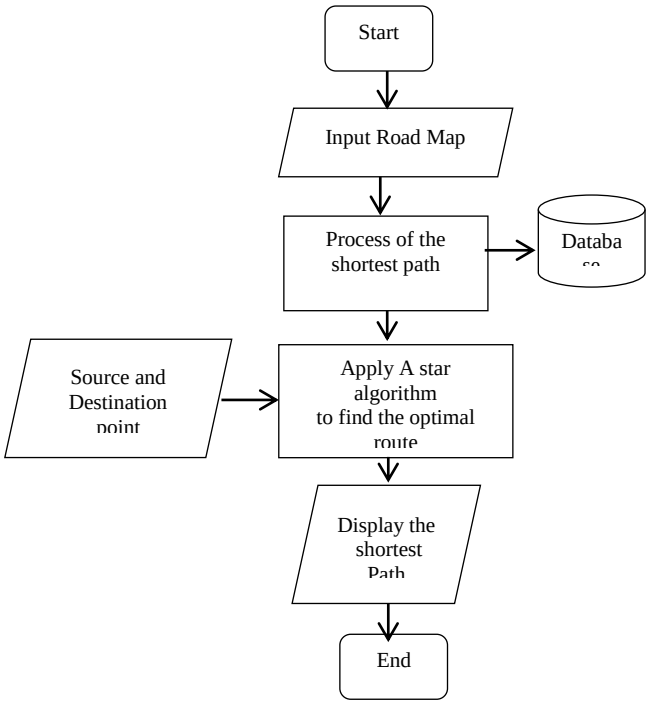


Figure 2. Process Flow of the System

2.8. Database Design

In this paper, user input road map nodes (Mandalay,Yangon,Taunggyi,Meitila,Kalaw,Phyu,Bago,etc.) information is saved in the database. Their coordinates are also saved to compute the heuristics estimate distances from source to destination. In this system ,there are used three tables in the database, Nodes table for storing nodes information and Distances table for storing distances (weight in graph) between nodes, and Transaction table for finding shortest path information. It is shown in Figure 3**.**

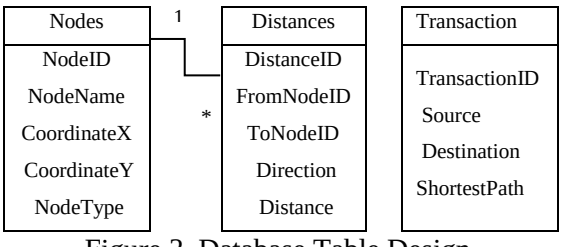


Figure 3. Database Table Design

2.9. Implementation of the Shortest Path Finding System

Road map input process by user is shown in Figure 4: In view of that, the road map of Taunggyi to Yangon as input by sample. User chooses any two cities of source city, Taunggyi and destination city, Yangon. User clicks the file menu from the main frame .When the user any road map as input to this process, if there is no road map, user input any new road map.

User must be double clicks in current map area to input the city or point. It also, links their distance. For that reason, numbers of cities are adding in database because of no limitation. As well as, the distance between of any two nodes can be change easily and it can remove the cities above way it can edit. Select source city and target city from the map is shown in Figure 5: To find the shortest path of desire cities by choosing "Search Shortest Path" under the "Algorithm Menu". User can input any two points or source city (Taunggyi) and destination (Yangon). The information given by user has to be calculated with A * algorithm that is generated by the system. During shortest path finding system, this will take a few seconds to wait for user there is a modification of the system.

In additionally, total distance of Taunggyi (source city) to Yangon (target city) is 422miles. The cities which shortest path passed is also displayed with their distances i.e. Taunggyi to Kalaw is 44.1 miles, Kalaw to Pyawbwe is 61.8 miles, Pyawbwe to Naypyidaw is 76.3 miles, Naypyidaw to Taungoo is 71.5miles, Taungoo to Phyu is 44.5 miles , Phyu to Bago is 79.9 miles and Bago to Yangon is 44.0 miles.

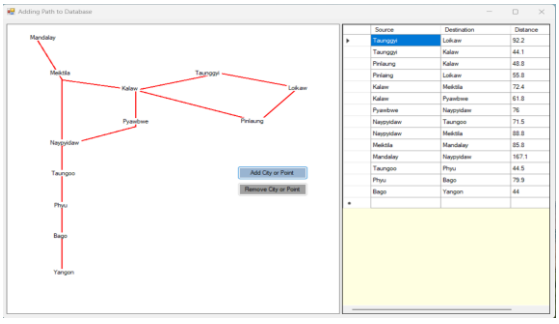


Figure 4. Road map input process by user

A * algorithm can be found shortest path even so many points. This method is helpful for visitor and Tourism. Furthermore, this algorithm is benefitted for bus driver because of decreasing the fuel. A * algorithm is the best algorithm for shortest path finding system. Output result of solution path is displayed in Table 3:

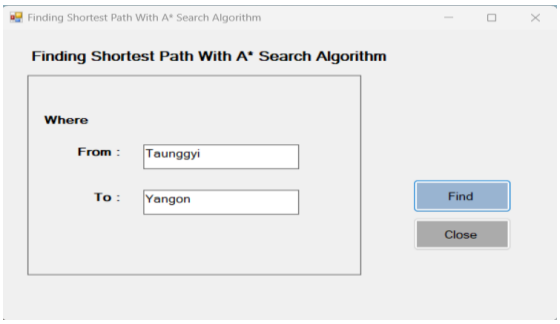


Figure 5. Select source and target

Table 3: Output result of the system

Result Path	Taunggyi-Kalaw-Pyawbwe-Naypydaw-Taunggo-Pyay-Bago-Yangon	
Result		
FROM	TO	DISTANCE (miles)
Taunggyi	Kalaw	44.1
Kalaw	Pyawbwe	61.8
Pyawbwe	Naypyidaw	76.3
Naypyidaw	Taungoo	71.5
Taungoo	Phyu	44.5
Phyu	Bago	79.9
Bago	Yangon	44
	Total Distances	422miles

The shortest distance is 422 miles as the following pattern with the shortest path from Taunggyi to Yangon.

“Taunggyi,Kalaw,Pyawbwe,Naypyidaw,Taungoo,Pyay,Bago,Yangon”

If user chooses other any two cities of source city,Mandalay and destination city,Loikaw then the optimal route is also displayed with their distances i.e. Mandalay to Meiktila is 85.8 miles, Meiktila to Kalaw is 72.4 miles, Kalaw to Pinlaung is 48.8 miles, Pinlaung to Loikaw is 55.8miles.The shortest distance is 262.8 miles as the pattern Mandalay-Meiltila-Kalaw-Pinlaung-Loikaw.

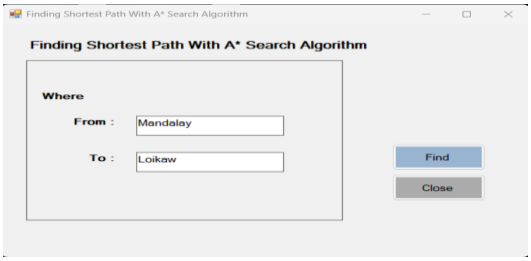


Figure 6. Select source and target

3. CONCLUSIONS

Travelling agencies system is now widely being used in many business areas. This system supports flexibility for users in the shortest path finding system using A * algorithm. The user can easily use for choosing of road from his road map. When the user gives the relation of the cities and distances between them, the shortest path of between cities can be found by this system. And then, because of the calculation of shortest path it can remove the difficulties of time and cost in travelling. Therefore the system saves cost and time consumption. Not only public users but also travelling agencies more easily for improve their business by using A * Algorithm. System will support the shortest path of traveling process before the travellers go to anywhere. It is use the mathematical and algorithm analysis field in the computerized system.

ACKNOWLEDGEMENTS

The author would like to express deepest gratitude to Dr. San San Tint, Pro Rector of University of Computer Studies (Mandalay) giving me helpful guidance, effective suggestion, support and encouragement.

REFERENCES

[1] Stuart Russell and Peter Norving, "Artificial Intelligence, A Modern Approach", Second Edition.

[2] https://www.guru99.com/graph-types-example.html

[3] Mariusz Gł ˛abowski, Bartosz Musznicki, Przemysław Nowak and Piotr Zwierzykowski (2013)," Efficiency Evaluation of Shortest Path Algorithms ", The Ninth Advanced International Conference on Telecommunications.

[4] Xiao Cui & Hao Shi (2011)" A*-based Pathfinding in Modern Computer Games", Vol.11, No.1.

[5] Nawaf Hazim Barnouti, Sinan Sameer Mahmood Al-Dabbagh, Mustafa Abdul Sahib Naser (2016) "Pathfinding in Strategy Games and Maze Solving Using A* Search Algorithm", Journal of Computer and Communications, Vol.4, No.11.

[6] Siau K (2018), "Education in the age of Artificial Intelligence: How will technology shape Learning? The global analyst", Vol.7, No.3, pp22-24.

Authors

Biography

Su Su Win was born in Nyaung Shwe in 1984 and achieved M.C.Sc degree in 2010 from University of Computer Studies (Loikaw). She works as a Lecturer in Faculty of Computer Science at the University of Computer Studies (Mandalay).