****
****

Proceedings of

National Journal of Parallel and Soft Computing

Volume 04, Issue 01

Organized by

University of Computer Studies, Yangon

Ministry of Science and Technology, Myanmar

July, 2025

EDITORIAL BOARD

REVIEWER BOARD

National Journal of Parallel and Soft Computing Volume 04, Issue 01

July, 2025

CONTENTS

****

Decision Making on Green Food Packaging for Reducing Environmental Impact
Hnin Wai Khaing, Myat Thiri Khine, Yu Yu Khaing 1-6

Imperceptible Image Steganography with Generative Adversarial Network (GAN)
Aye Mya, Nwe Thazin 7-12

Classification of Bank Marketing Data Using Support Vector Machine
Ei Ei Khin, Tin Tin Htar 13-18

The Car Insurance Claim Prediction System by using Machine Learning Algorithms on Apache Spark Platform
Thein Than Ko, Tin Zar Thaw 19-24

Secure Video Steganography by Using Knight Tour Algorithm and Least Significant Bit (LSB) Method
Aye Thandar Htun, Nwe Thazin 25-30

Investigation The Effect of Injury Severity Factors Based on Mutual Information Approach
Mya Thin Khaing, Yawai Tint 31-36

Detection and Classification of Paddy Leaf Disease based on the Combination of OTSU's Method and Support Vector Machine (SVM)
May Phyu Phyu Aung, Phyo Phyo Wai 37-42

Phishing URL Detection System using Ensemble Learning Methods
Theint Theint Win, Tin Zar Thaw 43-48

Classification of DDoS Attacks using Random Forest Algorithm
Su Su Naing, Khaing Khaing Wai 49-54

Facial Expression Classification by Using Local Binary Pattern and VGG16 Network
Aye Thant Thant Moe, Khant Kyawt Kyawt Theint 55-60

Strengthening Web Security for Input Validation Approach in PHP Web Development using Knuth-Morris-Pratt Pattern Matching Algorithm
Su Sandi Tun, Nilar Aye 61-66

Comparative Analysis of Feature-Based Phishing URL Classification Using Support Vector Machine
Nandar Lin, Phyo Phyo Wai 67-72

Transformer based Summarization on News
Aye Thinzar Ko, Dr. Lwin Nyein Thu 73-79

University Students' Stress Level Classification Using Random Forest Model
Thon Thant Thant Thaung, Hnin Pwint Phyu 80-85

Multi-Class News Classification using Fine-Tune BERT Model
Thiri Zaw, Nan Saw Kalayar 86-91

Resume Screening Job Classification System Using Three Machine Learning Algorithms
Nan Saw Kalayar, Chaw Yati Oo 92-97

Text Meets Ensemble Learning: Prediction House Prices with TF/IDF and XGBoost
Khine Cherry Htun, Yin Nyein Aye 98-103

Student Dropout Prediction System Using Three Machine Learning Algorithms
Su Myat Thwe, Nang Saw Kalayar 104-109

Rice Classification System Using Deep Learning
Soe Thinzar Mie, Paing Thwe Soe 110-115

Violence Detection System Using MobileNetV2 and Bidirectional GRU
Thet Hsu Myat, Darli Myint Aung 116-121

Travel Package Purchase Prediction System Using Ensemble Learning Algorithms
Thae Su Hlaing, Nan Saw Kalayar 122-127

Measuring Customers' Satisfaction on Restaurant Reviews Using Support Vector Machine
April Aung, Nan Saw Kalayar 128-133

Ingredient-based Recipe Recommendation System using Latent Dirichlet Allocation (LDA)
Ei Pyae Phyo Khin, Soe Soe Lwin 134-139

A Corpus-Based Method for Revealing Category Overlap in Research Paper
Kyawt Kay Thwe Htoo, Hsu Mon Kyi 140-145

Decision Making on Green Food Packaging for Reducing Environmental Impact

Hnin Wai Khaing, Myat Thiri Khine, Yu Yu Khaing
University of Computer Studies, Pathein
hninwaikhaing@ucspathein.edu.mm, myatthirikhine@ucspathein.edu.mm, yuyukhaing@ucspathein.edu.mm

Abstract

Green packaging development is a crucial tactic for minimizing the supply chain's environmental impact. In environment, there are having impact such as world warming, soil damage and ocean pollution. So, green packaging development is one of the important parts for green environment. Trade-offs are frequently made since it might be challenging for the organization to balance all the many environmental demands on packaging. Moreover, it is still required to develop the system that provide the organization for obtaining the suitable packaging to maintain the sustainability and green development. This paper implemented the multi-criteria decision support system based on various types of food package in order to loses the environmental impact. In this paper, the Fuzzy Technique for Order Preference by Similarity to the Ideal Solution (FTOPSIS) method is combined with Fuzzy-Analytical Hierarchy Process (FAHP) method to determine the rank based on the food packaging that is more suitable for the green environment. FAHP was employed to evaluate the fuzzy weight value for the green packaging based on criteria. FTOPSIS was employed to evaluate the rank value for green packaging based on the selective alternatives. Performance of this proposed system was compared with FTOPSIS method.

Keywords: Green packaging development, Green Supply Chain Management, Multi-Criteria Decision Making, FAHP, FTOPSIS

1. Introduction

Interest in global environmental conservation has increased recently. Green Supply Chain Management (GSCM) is a crucial factor in promoting sustainable development and reducing the effects of pollution and waste. Nowadays, organizations in both developing and developed country are involved in various GSCM activities in order to survive in the highly competitive market, attain customer reliance, improve brand value, and minimize ecological footprints. Green supply chain management (GSCM) activities include green reverse logistics, green packaging, green manufacturing, green design, and green marketing. This paper will focus on the green food packaging that it can be effective and safe the environment.

Green packaging development is one of the important part for obtaining green environment. Green packaging, also known as sustainable packaging or eco-friendly packaging, refers to packaging designs which have the lowest environmental impact possible. Food packaging contains harmful chemicals. So, it creates a lot of waste clogging our landfills and polluting our oceans. Using improper packaging can cause various negative side effect for the environment and for healthy living. Therefore, using proper packaging plays an importance role to preserve green environment when packaging the food.

This paper will support the organization for utilizing the suitable packaging based on food types to reduce the environmental impact. In addition, it also provides human to obtain the healthy lifestyle by avoiding the negative side effect of the environment. Moreover, it can also help business organization to develop effective a long-term plan for introducing sustainable green food packaging. In this paper, FAHP and FTOPSIS which are an effective tool for decision making will be applied for obtaining the suitable food packaging in order to support environment-friendly.

The rest of the paper is presented as follows. Section 1 describes the introduction of this paper. Section 2 discusses related works. Section 3

covers background theory and discusses about FAHP method and FTOPSIS method. Section 4 expresses the detail 2 explanation of the proposed system. Section 5 discusses the result and performance evaluation of the proposed method. In section 6, conclusion will be described.

2. Related Work

Nowadays, more and more people have been observed, realized and implemented for getting green environment. Multi-Criteria Decision Making (MCDM) are utilized in various decision support system such as supplier selection, contractor selection, etc. Yogesh Kumar Sharma, Alok Kumar Yadav [16] discussed on the projected replacement and reducing of plastic material for using in food packaging. Dikky Indrawan [8] applied AHP method that supports the purpose of choosing a new material for food packaging.

O. M. Causil; D.C.Morais [15] discussed the PROMETHEE II, PROMETHEE GDSS and the ROC to estimate the criteria weight values, which only allows Decision Maker to rank the attribute values. In Dang, W., Dang, V.-T., & Dang [7], Fuzzy Analytic Hierarchy Process (AHP) and fuzzy Technique for Order Preference by Similarity to Ideal Situation (TOPSIS) is presented to assess with difference ranking for evaluating many cities in a developing nation (Vietnam). Falak, J., Kunjan, M., Nagaraju, D., & Narayanan, S. [9] discussed to demonstrate how to choose the best continuous improvement approach using Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) and VIKOR. Abimbola H. Afolayan, Bolanle A. Ojokoh, and Adebayo O. Adetunmbi [5] proposed Multi-Criteria Decision Support Models with Fuzzy Analytic Hierarchy Process for Performance Analysis in Contractor Selection.

Meysam Shaverdi, Mahsa Akbari, Sajad Emamipour [14] proposed FAHP approach that performs well and can be applied to many real-world issues. In an effort to improve the quality of the information and ranking, FAHP has chosen to calculate the relative weights of the various selection criteria. It demonstrates the potential utility and potency of the suggested FAHP model for choosing the web browsers. Ananna Paul, Nagesh Shukla, [3] presented Multi-criteria decision-making and sustainable supply chain management Because of its computational capabilities, MCDM methods have been extensively used in the field of sustainable supply chain management (SSCM). Abimbola H. Afolayan, Bolanle A. Ojokoh [2] discussed about the performance analysis for MCDM model. Katrin Molina-Besch [13] proposed criteria for prioritizing food GPD based on an in-depth analysis of food packaging development. Also, it offers recommendations for enhancing packaging in the most crucial regions. It can assist food businesses in creating packaging options that lessen the environmental effect throughout the food supply chain.

Atef M. Ghaleb , Husam Kaid , Ali Alsamhan, Syed Hammad Mian, and Lotfi Hidri [4] discussed a variety of MCDM approaches are used in the manufacturing process selection. Also, the VIKOR and TOPSIS methodologies were more practical for choosing manufacturing procedures that would allow for agility during the decision-making process.

3. Background Theory

The set of techniques known as multiple criteria decision making (MCDM) analyze several, competing criteria when making decisions. It consists of forming a global preference relation for a set of alternatives evaluated using several criteria. It takes the best actions from a set of alternatives, each of which is assessed against multiple and often confliction criteria. Multi-criteria decision-making (MCDM) have been used for selecting better alternatives in various decision making problems [3].

There are several MCDM techniques. ANP (Analytic Network Process), AHP (Analytical Hierarchy Process), FAHP (Fuzzy Analytic Hierarchy Process), FTOPSIS (Fuzzy Technique for Order Preference by Similarity of an Ideal Solution) are mostly used among MCDM methods [15]. In order to make workable decisions in challenging and unclear circumstances, fuzzy set theory is frequently supplemented with traditional MCDM methods.

3.1. Fuzzy Analytic Hierarchy Process (FAHP)

In multi-criteria decision making, the AHP method has been widely applied. It is a popular, well-known, and simple way for making multi-

criteria decisions. AHP treats alternatives, which simplifies difficult decision-making issues. Fuzzy Analytic Hierarchy Process is a method of Analytic Hierarchy Process (AHP) developed with fuzzy logic theory. FAHP helps decision-makers in comprehending the elements' ultimate significance and their level of uncertainty [2]. The weight scores for each criteria are determined using the Fuzzy Analytical Hierarchy Processing (FAHP) method [7].

The following steps have been provided to explain FAHP:

Step 1: With the aid of the relative relevance scale, determine performance criteria.

Step 2: The pairwise comparison matrix of criteria.

Step 3: building the fuzzy pairwise comparison criteria matrix.

$$A = \begin{bmatrix} \tilde{a_{11}} & \tilde{a_{12}} & \dots & \tilde{a_{1n}} \\ \tilde{a_{21}} & \tilde{a_{22}} & \dots & \tilde{a_{2n}} \\ \tilde{a_{m1}} & \tilde{a_{m2}} & \dots & \tilde{a_{mn}} \end{bmatrix} \quad (1)$$

Step 4: Constructing the geometric mean value

$$g_i = \left(\pi_{j=1}^{n} a_{ij}\right)^{1/n} \quad (2)$$

Step 5: Determination the fuzzy weight of criteria

$$w_i = g_i * (g_1 + g_2 + \cdots + g_n)^{-1} \quad (3)$$

Where, g_i is the geometric mean value, w_i is the fuzzy weight of criteria, a_{ij} is the decision maker's perference for ccriterian over criterion in i items of Traigular Fuzzy Number (TFN), l_{ij} is lower limit , m_{ij} is medium limit and u_{ij} is upper limit

3.2. Fuzzy Technique for Order Preference by Similarity of an Ideal Solution (FTOPSIS)

The FTOPSIS approach can be used to evaluate alternatives based on a variety of criteria. The FTOPSIS method is a well-liked approach for handling decision-making issues [7]. An alternative selection concept utilized in multi-criteria decision-making processes is called FTOPSIS. Utilizing FTOPSIS, alternatives are ranked.

The following steps have been provided to explain FTOPSIS method:

Step 1: Choose the proper linguistic factors to use when comparing options based on a set of criteria.

Step 2: The pairwise comparison matrix of possibilities.

Step 3: Establish the fuzzy decision matrix as:

$$P = \begin{bmatrix} \tilde{p_{11}} & \tilde{p_{12}} & \dots & \tilde{p_{1n}} \\ \tilde{p_{21}} & \tilde{p_{22}} & \dots & \tilde{p_{2n}} \\ \tilde{p_{m1}} & \tilde{p_{m2}} & \dots & \tilde{p_{mn}} \end{bmatrix} \quad (4)$$

Step 4: Calculate the normalized fuzzy decision matrix

$$n_{ij} = \left[\frac{a_{ij}}{c_j}, \frac{b_{ij}}{c_j}, \frac{c_{ij}}{c_j}\right]; c_j = max\{c_{ij}\} \quad (5)$$

$$n_{ij}^{-} = \left[\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{a_{ij}}\right]; a_j^- = \min\{a_{ij}\} \quad (6)$$

Step 5: Create a fuzzy decision matrix that is weighted and normalized.

$$z_{ij} = n_{ij} * w_j \quad (7)$$

Step 6: the Fuzzy Positive Ideal Solution (FPIS) and the Fuzzy Negative Ideal Solution (FNIS)

$$\text{FPIS}(A^*) = (Z_1^*, Z_2^*, Z_3^*, \dots, Z_n^*) \quad (8)$$

$$\text{FPIS}(A^-) = (Z_1^-, Z_2^-, Z_3^-, \dots, Z_n^-) \quad (9)$$

Step 7: Find the separation between each alternative and FPIS and FNIS as

$$d_i^* = \sum_{j=1}^{n} d_z\left(z_{ij} \times Z_j^*\right); \text{ i=1, 2, 3, ...m} \quad (10)$$

$$d_i^- = \sum_{j=1}^{n} d_z\left(z_{ij} \times Z_j^-\right); \text{ i=1, 2, 3, ...m} \quad (11)$$

Step 8: For each option, calculate the closeness coefficient. The distances to the FPIS and FNIS used to rank the alternatives are represented by the closeness coefficient.

$$CC_i = \frac{d_i^-}{d_i^- + d_i^*} \quad (12)$$

Step 9: Sort the options in order of closest to best, using the proximity coefficient.

Where, p_{ij} is the decision maker's perference for ccriterian over criterion in i items of Traigular Fuzzy Number (TFN), n_{ij} is the normalized fuzzy decision matrix, z_{ij} is the weight normalized fuzzy decision matrix, w_j is the fuzzy weight of evaluating criteria, CC_i is the closeness coefficient

for each alternative, a_{ij} is lower limit, b_{ij} is medium limit and c_{ij} is upper limit.

4. Proposed System

The entire flow of the proposed system is shown in figure 1. This system implements the decision making on green food packaging for reducing environmental impact using MCDM.

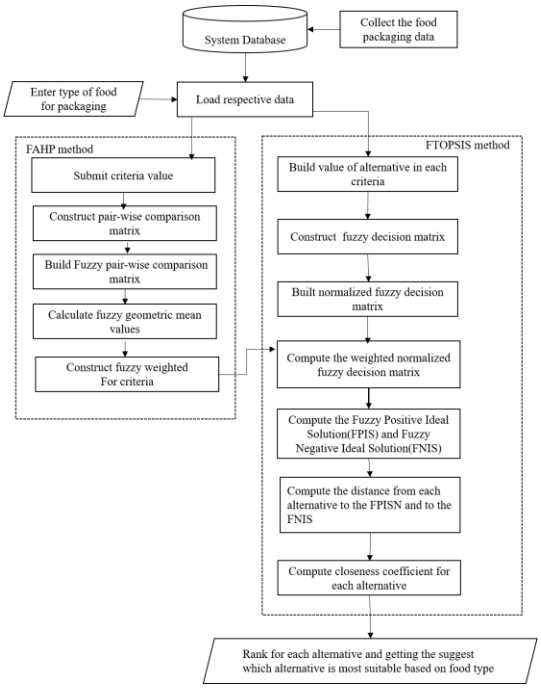


Figure 1. Proposed System

Food packaging data is stored in the database. Food type is required to choose for determining the rank of materials. Moreover, the desired criterias such as type of material, properties, environmental profile are required to select. Then, the respective data is loaded from the database based on the food type. The loaded data will be used for calculation in the FAHP and FTOPSIS methods. There are five steps in FAHP method that submit criteria value, construct pair-wise comparison matrix, build fuzzy pair-wise comparison matrix, calculate fuzzy geometric mean values and construct fuzzy weight. This part will generate the fuzzy weight value for each criteria that will be used into the weighted normalized fuzzy decision matrix for FTOPSIS.

There are eight steps in FTOPSIS. First step is to build value of alternative in each criteria taking from load respective data. Then, construct fuzzy decision matrix and build normalized fuzzy decision matrix will be made. After that, fuzzy weight value of FAHP is taken to calculate the weighted normalized fuzzy decision matrix. The Fuzzy Positive Ideal Solution(FPIS) and Fuzzy Negative Ideal Solution(FNIS) are computed and then the distance from each alternative to the FPISN and to the FNIS will be computed. Closeness coefficient for each alternative is computed to get the ranking for each alternative. Finally, the proposed system will output the ranking list of alternative and suggest which alternative is most suitable for environment.

5. Performance Evaluation

The performance of the FAHP and FTOPSIS models is evaluated using Mean, Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). The better the model fits, the lower its MSE, RMSE, and MAE values, and the better it fits, the higher its accuracy. By contrasting the closeness coefficients produced by the suggested FAHP and FTOPSIS models, the performance analysis is carried out. The following performance measures will be used to show performance:

$$\text{Mean: } X = \frac{\sum_{i=1}^{n} X_i}{n} \tag{13}$$

Where the measurement's outcome, the arithmetic mean (i^{th}), and the number of observations, n, are all present.

$$\text{MSE} = \frac{\sum_{i=1}^{n} (X_i - X_i)^2}{n} \tag{14}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \overline{X_i})^2}{n}} \tag{15}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n} |(X_i - \overline{X_i})^2|}{n} \tag{16}$$

Fifteen material types and fifty food types are used in this system. Data information is collected from the food market in Myanmar. Properties of material type is study from the organization such as warmheart [14], matmatch [15] and ncbi.nlm.nih.gov [16]. Materials data are Glass, Aluminum, Tinplate, Tin-free Steel, polyolefin, Metallized Films, Polyvinyl Chloride, Polystyrene, Polyamide, Ethylene Vinyl Alcohol, Co-extrusions, Kraft paper, Sulfate paper, Greaseproof and white board. In Table 1 and 2, four material data were shown for food type of dried fruit.

Table 1. Ranking result of proposed method

Package Material	Closeness Coefficient	Ranking
Glass	**0.907513**	**1**
Polyolefin	0.760322	2
Tin-free steel	0.590552	3
Tinplate	0.105176	4

Table 2. Ranking result of FTOPSIS method

Package Material	Closeness Coefficient	Ranking
Glass	**0.905815**	**1**
Polyolefin	0.775681	2
Tin-free steel	0.602813	3
Tinplate	0.092975	4

The performance evaluation is tested on dried fruit type. The rank of material based on selective food type using proposed method is shown in table 1. In this table, closeness coefficient value of glass material is 0.893011, Polyolefin is 0.770943, Tin-free steel is 0.660409 and Tinplate is 0.087232. The value of glass material is greater than other. So, glass material is rank 1. Table 2 describe the result of dried fruits type using FTOPSIS method. Closeness coefficient value of glass material is 0.891705, Polyolefin is 0.784139, Tin-free steel is 0.671668 and Tinplate is 0.076866. The value of glass material is greater than other. So, glass material is rank 1.

In this two table, the rank is same but closeness coefficient value is difference. When the closeness coefficient values of proposed method are compared with FTOPSIS method, the glass value of proposed method is greater than FTOPSIS method. So, the proposed method is better fit.

Table 3. Performance analysis of proposed method and FTOPSIS method

Performance Metrics	Proposed method	FTOPSIS method
MSE	0.091219	0.095334
RMSE	0.302024	0.308762
MAE	0.302024	0. 308762

The performance analysis of proposed method and FTOPSIS method is described in Table 3. In this table, the proposed method and FTOPSIS method approximately generated the difference result for MSE, RMSE, MAE. Performance values of proposed method are MSE=0.091219, RMSE=0.302024 and MAE=0.302024. Performance values of FTOPSIS method are MSE=0.095334, RMSE=0.308762 and MAE=0.308762. The performance of proposed method is better than FTOPSIS method because the performance values of proposed method is lower than FTOPSIS method.

6. Conclusion

Green food packaging reduces environmental impact and stable business organization to develop and effectively implement a long-term strategy. This paper provided for obtaining the suitable green food packaging by using MCDM. In this paper, combination of the two methods (FAHP and FTOPSIS) were used to choose the suitable package according to the rank of material. FAHP method was employed to calculate the fuzzy weight of criteria. FTOPSIS was used to rank the alternative. Ranking list of the material type can be obtained and the most suitable type of package can also be perceived for green environment. In addition, it also provided organization to use suitable food packaging for diminution the waste of environment.

References

[1] Abdallah, A. B., & Al-Ghwayeen, W. S. (2019). “Green supply chain management and business performance”. Business Process Management Journal.

[2] Abimbola H. Afolayan, Bolanle A. Ojokoh, “Performance Analysis of Fuzzy Analytic Hierarchy Process Multi-Criteria Decision Support Models for Contractor Selection”, Scientific African (2020).

[3] Ananna Paul, Nagesh Shukla, “Sustainable Supply Chain Management and Multi-Criteria Decision-Making Methods”, School of Information, Systems and Modelling, University of Technology Sydney, 2021.

[4] Atef M. Ghaleb , 1 Husam Kaid , 1 Ali Alsamhan,1 Syed Hammad Mian,2,3 and Lotfi Hidri 1 “Assessment and Comparison of Various MCDM Approaches in the Selection of Manufacturing Process”, Advances in Materials Science and Engineering ,2020

[5] Abimbola H. Afolayan, Bolanle A. Ojokoh, and Adebayo O. Adetunmbi, “Performance Analysis of Fuzzy Analytic Hierarchy Process Multi-Criteria

Decision Support Models for Contractor Selection", Scientific African, 2020.
[6] Bhattacharya, A., Mohapatra, P., Kumar, V., (2013). "Green supply chain performance measurement using fuzzy ANP-based balanced scorecard: A collaborative decision-making approach". Production Planning & Control, 25(8), 698–714.
[7] Dang, W., Dang, V.-T., & Dang, (2019). "An Integrated Fuzzy AHP and Fuzzy TOPSIS Approach to Assess Sustainable Urban Development in an Emerging Economy". International Journal of Environmental Research and Public Health, 16(16), 2902.
[8] Dikky Indrawan "Replacing Plastic: An Assessment of New Material for Food Production Package to Re-Engineer Packaging Industry Based on Multi-Criteria Analyses",2019
[9] Falak, J., Kunjan, M., Nagaraju, D., & Narayanan, S. (2020). "Evaluation of continuous improvement techniques using hybrid MCDM technique under fuzzy environment". Materials Today: Proceedings, 22, 1295–1305.
[10] https://warmheartworldwide.org/climatechange/?gclid=CjwKCAjwvNaYBhA3EiwACgndgkMMLU2IohrHVgsLNEamaccMUuSlbqfWN2hnqtmvJEqAWwfn7MfwtBoC6OEQAvD_BwE
[11] https://matmatch.com/learn/material/materials-used-in-food-packaging
[12] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7664184/
[13] Katrin Molina-Besch, "Prioritization guidelines for green food packaging development", British Food Journal, 2016.
[14] Meysam Shaverdi, Mahsa Akbari, Sajad Emamipour "Using Fuzzy Multi Criteria Decision Making Approach for Ranking The Web Browsers", International Journal of Economics and Management Sciences, 2012.
[15] O. M. Causil; D.C.Morais "A Multi-criteria Group Decision-making Model for Selecting a Perishable Food Packaging System Using an Outranking Method",2021
[16] Yogesh Kumar Sharma, Alok Kumar Yadav "Ranking the Success Factors to Improve Safety and Security in Sustainable Food Supply Chain Management Using Fuzzy AHP",2017

Imperceptible Image Steganography with Generative Adversarial Network (GAN)

Aye Mya, Nwe Thazin
University of Computer Studies (Taunggyi)
ayemya@ucstgi.edu.mm, nwethazin@ucstgi.edu.mm

Abstract

Image steganography involves concealing messages within pictures, with the goal of hiding the presence of the message rather than preventing adversaries from reading it. Recent advances in the field of Deep Learning have led to the discovery of an unsupervised learning network called the Generative Adversarial Network "GAN", which can be used for intelligent steganography. This paper presents a simple GAN-base image steganography system to hide arbitrary binary data in images while optimizing the perceptual quality of the resulting stego images. The system was evaluated using two datasets, Div2k and Custom, and the results indicate that the steganographic image quality is high across various variants.

1. Introduction

Steganography dates back thousands of years to ancient Greece and has a fascinating history, where physical materials like invisible ink were used to conceal secret messages. Today, steganography relies on digital files like images to hide messages. In modern steganography, the sender encodes a message within an image, and the receiver uses a decoder to extract it. This process is typically imperceptible to anyone other than the sender and receiver of the message [1].

The field of image classification has experienced significant growth in digital media technologies, and has attracted considerable attention from the research community. Initially, traditional methods were employed to conceal information in cover images using the pixel values of the images. However, traditional approaches to image steganography are restricted by their embedding capabilities and security concerns.

A new approach to image steganography is advanced with the rise of deep learning over the past decade [2]. Generative adversarial networks "GANs" of deep learning methods have been developed in image steganography. These methods increase concealability and security compared to traditional methods [3].

This study focuses on utilizing Generative Adversarial Network "GAN" for image steganography. The primary objective is to embed secret messages into images with enhanced image quality and greater embedding capacity.

2. Motivation

Steganography is a useful means of secret communication in many fields. In medicine, for example, steganography can be utilized to conceal personal patient information, such as biometric data, within images like X-rays or MRIs. Similarly, in the media industry, steganography can be employed to embed copyright data and access control systems for distributing digital content on the Internet. In each of these scenarios, it is crucial to maximize the amount of data embedded while maintaining undetectability, and the receiver must be able to recover the data without any loss. This paper, along with other works in steganography, aims to achieve these two objectives, and proposes a new class of models for image steganography that can effectively fulfill both goals.

3. Related Works

To explore into steganography research, it is essential to thoroughly examine the current literature. This section undertakes such an investigation and discusses various types of steganography.

The conventional method for performing image steganography is to use the Least Significant Bits "LSB" substitution method. In [4], the authors employed a variation of the Least

Significant Bit "LSB" method specifically designed for RGB images. This variant involves dividing the cover image into its red, green, and blue channels, and then embedding the secret message in each channel according to a 2:2:4 ratio for the red, green, and blue planes, respectively.

Duan et al. proposed a CNN-based method for image steganography in [5], which uses a U-Net based encoder-decoder architecture for hiding and a CNN with six layers for extraction. The U-Net is modified to accept input images of size 256 × 256 and six channels, obtained by concatenating the cover and secret images. Van et al. also use a U-Net based approach in [6], but with separate networks for hiding (H-net) and revealing (R-net), using batch normalization and ReLU activation. They concatenate the cover and secret images before feeding them to the network and use two optimization losses based on "SSIM" and "MSE" to improve performance.

In 2014, Goodfellow et al. [7] introduced General Adversarial Networks "GANs" as a variant of deep CNNs. GANs employ game theory to train a generative model through an adversarial process, which has made them popular for image generation tasks. Numerous variants of GAN have been proposed since then, enhancing its capabilities and adaptability for generating synthetic images.

Yang et al. [8] introduced an image steganography approach based on GAN, which comprises three modules: Generator, Embedding Simulator, and Discriminator. The authors named the architecture UT-GAN because they utilized the U-Net based design in the generator network. The U-Net based generator produces a probability map "P" for the cover image "C" because of its excellent pixel-wise segmentation performance. The U-Net in this model has both contracting and expanding convolutional layers, with the number of filters increasing for convolution layers and decreasing for deconvolution layers. The prior research has shown that GANs have delivered remarkable results in image generation.

4. System Architecture

This system uses a deep method to embed the arbitrary bit vector into a cover image. This system uses a GAN composed of two neural networks. In our model, the fundamental building block of computation is a convolutional block. The architecture is comprised of a convolutional layer with a stride of 1 and a kernel size of 3, followed by a ReLU activation, and a batch normalization layer.

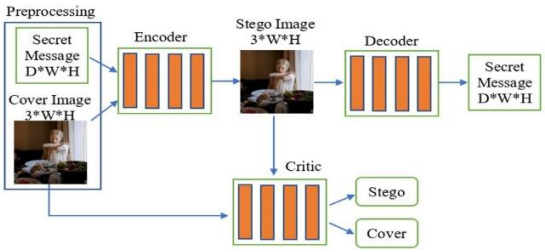


Figure 1. System Architecture

The system architecture is shown in figure 1, which consists of three neural networks and preprocessing module. The three neural networks are Encoder, Decoder, and Critic networks. The Encoder that takes a cover image and a data tensor as input and generates a steganographic image as output. The Decoder that takes a steganographic image as input and attempts to recover the data tensor. The Critical network that attempts to distinguish between cover and steganographic image. The cover image and secret message need to be preprocessed in the preprocessing module to obtain appropriate input for the encoder network.

4.1. Data Preprocessing

Prior to training and testing, image and message preprocessing are required in this system.

4.1.1. Image Preprocessing

Image preprocessing is done using standard data augmentation to avoid the problem of over fitting in this system. There are four steps of data augmentation shown in figure 2.

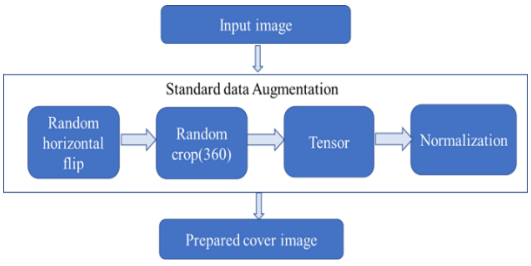


Figure 2. Image Preprocessing

The input image was random flip horizontally, random crop, change to the tensor value and followed the normalization process to improve the model accuracy.

4.1.2. Message Preprocessing

In message pre-processing process, there are three steps before training shown in figure 3. In this phase, the user's input message is first converted into a byte array, which is then transformed into (0,1) bits. These bits are then fed into TensorFlow (N×H×W) and converted into 32 bits. Once these steps are finished, a Data tensor is obtained that can be embedded into the cover image.

Figure 3. Message Preprocessing

4.2. Encoder

The encoder network's job is to take the cover image C and add the text which the user wants to hide. This text has been converted to a binary data tensor "M" (D*W*H). Where "D" is the number of bits want to hide in each pixel of the cover image and "W" and "H" describes the width and height of the cover image. The cover image is converted to a tensor "a" using convolutional block as shown in equation (1).

$$a = \mathrm{Cov}_{3\to 32}(C) \quad (1)$$

The depth "D" of the message "M" is concatenated to channel feature of cover frame "a" into (32+D) and the result is processed with a convolutional block as shown in equation (2).

$$b = \mathrm{Cov}_{32+D\to 32}(\mathrm{cat}(a, M)) \quad (2)$$

After that, additional convolution blocks are applied to this combined tensor "b", which perform further processing before generating the final steganographic image. This image will appear similar to the cover image "C" in terms of visual appearance, resolution, and depth.

4.3. Decoder

The decoder is responsible for extracting messages and takes the stego image frame, which is the encoder output, as input. Since the function performed by the decoder and encoder is mutually inverse, their network structures are quite similar. The decoder architecture comprises of 4 stages. The first 3 stages include a convolutional layer with 3 × 3 kernels, with a step of 1 and padding of 1, followed by a Leaky ReLU activation layer and batch normalization layer. The final layer is a convolutional layer with a kernel size of 3 × 3, step of 1, padding of 1, and the number of filters equivalent to the data depth.

4.4. Critic

The Critic functions as steganalysis, taking both cover and steganographic images as inputs. It consists of three convolutional blocks and a final convolutional layer, with a mean pooling layer applied to obtain scores for cover and stego frames. The Critic trains to differentiate between cover and stego images, providing feedback for the Encoder and producing more realistic images. Through alternating iterative training, the Encoder's performance improves, and stego images become indistinguishable to the Critic.

4.5. Loss Function

This system iteratively optimizes the Encoder-Decoder network and Critic Network. To optimize the Encoder-Decoder network, this system associates three losses as shown in the equations (3-6) below:

- In the Decoder network (cross entropy loss function) is used to evaluate the decoding accuracy.

$$L_d = \mathrm{crossentropy}(\mathrm{decoder}(S), M) \quad (3)$$

- In the Encoder (mean square error) is used to measure the similarity between the steganographic image and the cover image.

$$L_e = \frac{1}{3\times W\times H}(C - \mathrm{Encoder}(C, M)) \quad (4)$$

- The realness of the steganographic image is used (generated score) from Critic network.

$$L_c = \mathrm{Critic}(S) \quad (5)$$

The objective of the system is to minimize the total loss.

$$\text{minimize } L_d + L_e + Lc \quad (6)$$

5. Training

During the training process, the Encoder-Decoder network and Critic network are trained simultaneously through adversarial learning. The goal is to optimize their performance and update them accordingly. The Encoder-Decoder networks pair aims to generate steganographic images that preserve the original image quality while embedding secret data. On the other hand, the Critic network is trained to accurately distinguish between the original and generated images. The training procedure is presented in Algorithm 1 and 2, which outlines the step-by-step process for updating the Encoder-Decoder and Critic networks.

During each epoch, update the model parameters by running Algorithm 2 followed by Algorithm 1 for all training set. This system is trained for 32 epochs using the Adam optimizer with a learning rate of $1e^{-4}$. The critic weights are limited to a range of [-0.1, 0.1], and the gradient norm is limited to 0.25.

Algorithm 1: Encoder-Decoder Network

1. Select a cover image C of shape (3, W, H)
2. Generate a data tensor M of shape (D, W, H)
3. Create a steganographic image: S = Encoder (C, M)
4. Recover the data tensor: M = Decoder(S)
5. Compute decoder loss: L_d = *CrossEntropy (Decoder(S), M)*
6. Compute the mean squared loss for Encoder:
$$L_e = \frac{1}{3 \times W \times H}(C - Encoder(C, M))$$
7. Compute Critic loss: L_c = *Critic(S)*
8. Update the encoder and decoder to minimize total loss:
$$\text{minimize } L_d + L_e + L_c$$

Algorithm 2: Critic Network

1. Select a cover image C of shape (3*W*H)
2. Generate a data tensor M of shape (D*W*H)
3. Create a steganographic image: S = Encoder (C, M)
4. Compute the Wasserstein loss: L_c = *Critic(C) - Critic(S)*
5. Update the critic to minimize the loss

Once the Critic model is trained, the optimization focus is shifted to the Encoder network. The objective is to reduce the score given to cover images while increasing the score given to steganographic images. This process compels the Encoder to generate more realistic steganographic images that are difficult to differentiate from cover images.

6. Evaluation Metrics

The proposed system is evaluated along three metrics. The amount of data (capacity) that can be hidden in an image payload capacity "BPP", similarity between cover and stego image "SSIM", picture (distortion) "PSNR" and "MSE" are used to measure the performance of this system. Below are their equations.

$$\text{BPP} = \frac{\text{number of bits in message}}{\text{total number of image pixel}} \quad (7)$$

$$\text{MSE} = (m \times n)^{-1} \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i,j) - K(i,j)]^2 \quad (8)$$

Where, "m, n" is the dimensions of cover the image and "I" is the original Image and "K" is stego-image. After calculating "MSE", its value is used in the calculation of "PSNR".

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \quad (9)$$

Where the "MAX^2" represents the maximum possible difference in the numerical representation of each pixel.

$$\text{SSIM} = \frac{(2\mu_x\mu_y + (k_1 L)^2)(2\sigma_{xy} + (k_2 L)^2)}{((\mu_x)^2 + (\mu_y)^2 + (k_1 L)^2)((\sigma_x)^2 + (\sigma_y)^2 + (k_2 L)^2)} \quad (10)$$

The default configuration for "SSIM" uses k_1=0.01, k_2= 0.03 and return values in the range [-1.0,1.0] where 1.0 indicates the images are identical. The "SSIM" can be computed using the means "μ_x" and "μ_y", variances, $(\sigma_x)^2$ and $(\sigma_y)^2$, and covariance "σ_{xy}" of the images.

7. Result and Analysis

This system trains and tests using the Div2k dataset and a custom dataset. The Div2k dataset has 1000 images, with 800 for training and 100 images for validation and 100 images for testing [9]. The custom dataset has 2200 images, with 2000 for training and 200 for testing [10]. All images are resized to 1024x1024. Secret information "M" is randomly sampled from a uniform distribution and embedded in the images with an embedding rate (data_depth) ranging from 1bpp to 4bpp, equivalent to 2KB to 8KB of arbitrary data. The model is trained with a batch size of 4 and for 32 epochs. To decrease the training time of our model, Google Colab was used. The model

was trained on the Div2k dataset for 32 epochs, with each epoch taking around 5 minutes. However, for the custom dataset, each epoch took approximately 8 minutes.

Table 1. The image quality metrics with relative payload for each dataset

Data depth	Div2k Dataset				
	MSE	PSNR	SSIM	Acc	BPP
1bpp	0.000	39.236	0.939	0.983	0.968
2bpp	0.001	37.092	0.915	0.957	1.829
3bpp	0.001	36.137	0.829	0.916	2.49
4bpp	0.001	37.109	0.865	0.805	2.437
Data depth	Custom Dataset				
	MSE	PSNR	SSIM	Acc	BPP
1bpp	0.000	44.672	0.939	0.999	0.999
2bpp	0.000	42.928	0.942	0.998	1.993
3bpp	0.000	39.830	0.829	0.965	2.789
4bpp	0.001	38.539	0.869	0.947	3.573

Table (1) presents a performance comparison of the two datasets at four distinct embedding rates (1bpp, 2bpp, 3bpp, 4bpp). It is important to note that there is a tradeoff between relative payload and image quality metrics. As the embedding rate increases, the quality of the image decreases. Nevertheless, the number of data bits that can be concealed in each pixel of the image increases with the embedding rate.

Of the four embedding rates in the div2k dataset, the lowest "MSE" is observed at 1bpp where almost one bit of data can be embedded in each pixel of the cover image, though with lower embedding capacity compared to the other rates. All four models in the div2k dataset experience minimal image corruption post-embedding, as indicated by the "SSIM" distance being close to 1. This suggests that the cover images and stego images generated by the encoder from these models are virtually indistinguishable to the human eye, making it difficult to detect the presence of steganographic data.

In the custom dataset, the performance of the four models well than div2k dataset. Due to the size of train dataset size increase, it shows good embedding quality and capacity than div2k. The results of this experiment are presented in figure (4). In this test result, div2k dataset can cover a maximum of (2bpp) in image and a custom dataset can cover a maximum of (3bpp) in image. The "SSIM" value is close to 1. The stego image and cover image are identical because of the damage is less after embedding process.

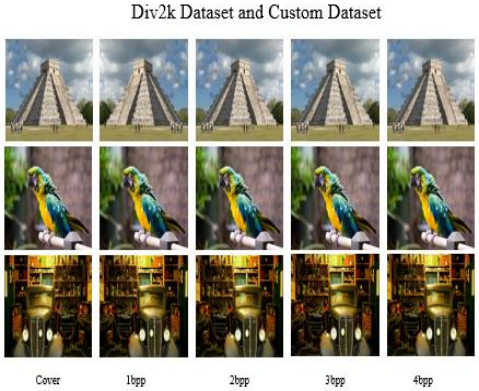


Figure 4. Given some cover images from Div2k and custom datasets and the stego image with embedding rate of 1bpp, 2bpp, 3bpp, 4bpp

8. Conclusion

This paper presents a GAN-based image steganography system. The system utilizes a deep learning approach to enhance both the quantitative and qualitative performance of image steganography, as demonstrated with the use of two datasets. Out of the two datasets used in this system, the custom dataset has better embedding quality and embedding capacity, likely due to its larger training dataset size. The custom dataset allows for a maximum embedding rate of 3bpp, while the div2k dataset supports only up to 2bpp, with minimal damage after embedding, as indicated by a close to 1 "SSIM" value. The system currently hides text messages in images only, but can be improved to embed data in different digital carriers like audio and video.

References

[1] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K.- H. Jung, "Image steganography in spatial domain: A survey," Signal Processing: Image Communication, vol. 65, pp. 46 – 66, 2018.

[2] Manglem Singh, K. S. Birendra Singh and L. Shyam Sundar Singh," Hiding Encrypted Message in the Features of Images", IJCSNS, VOL.7 No.4, April 2007.

[3] S. Baluja, ''Hiding images in plain sight: Deep steganography,'' in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 2069–2079.

[4] A. Singh and H. Singh, ''An improved LSB based image steganography technique for RGB images,''

In 2015 IEEE International Conference on Electrical, Computer and Communication Technologies, pp. 1-4. IEEE, 2015.
[5] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, ‘‘Reversible image steganography scheme based on a U-Net structure,’’ IEEE Access, vol. 7, pp. 9314–9323, 2019
[6] T. P. Van, T. H. Dinh, and T. M. Thanh, ‘‘Simultaneous convolutional neural network for highly efficient image steganography,’’ in Proc. 19th Int. Symp. Commun. Inf. Technol. (ISCIT), Sep. 2019, pp. 410–415
[7] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." *Communications of the ACM* 63, no. 11 (2020): 139-144.
[8] Yang, Jianhua, Kai Liu, Xiangui Kang, Edward K. Wong, and Yun-Qing Shi. "Spatial image steganography based on generative adversarial network." *arXiv preprint arXiv:1804.07939* (2018).
[9] https://data.vision.ee.ethz.ch/cvl/DIV2K
[10] https://unsplash.com

Classification of Bank Marketing Data Using Support Vector Machine

Ei Ei Khin, Tin Tin Htar
University of Computer Studies, Yangon, Myanmar
eieikhin@ucsy.edu.mm, tintinhtar@ucsy.edu.mm

Abstract

Nowadays, banking system plays an important role of financial sectors all over the world. The more accurate predictive modelling system is required for their services or products in the banking industry. Bank workers can make those predictive models with manually, but this process takes long time and lots of man-hours. For these reasons, machine learning techniques are useful to predict the outcomes with huge amounts of data. Classification is one of the most important techniques to analyse and to predict the new data. This system will implement the classification of bank marketing data using support vector machine (SVM) to predict the probability of the customers' subscription to the term deposit whether subscribe or not. Support Vector Machine (SVM) is a supervised learning model used for classification and prediction of data. The purpose of this system is to predict the customers' response using bank marketing data. The accuracy of this system is measured using confusion matrix (Precision, Recall, F-Measure).

Keywords— Classification, Bank Marketing Data, Support Vector Machine (SVM), Principal Component Analysis

1. Introduction

Data is widely used in several areas and the ones who holds a huge amount of data will be the one who holds the sources of information that is necessary for their business development. The banking sector is one of the sectors that holds the customers data which can make decisions on the services of banking systems. In [10], bank deposit is one of the bank products that saves the customers to use to hold an amount of money at a bank for a specific length of time. In return, the financial institution will pay the customers the relevant amount of interest, based on how much they choose to deposit and for how long.

In [1], deposit in banking system is a fixed-term investment and the main source of revenue for banks. Many banks offer different types of accounts to attract customers willing to deposit their funds. A bank can increase the number of subscribers to term deposit and can collect huge amounts of customer data through effective marketing. Bank marketing campaign can be carried out or launched in various ways using telephone, social media, emails, short message services, blogging, and others. The purpose of bank marketing campaign is to meet the targeted needs of the customers to satisfy the bank's product.

Banking industry requires more accurate predicting modeling system because of various challenges offering products or services. Data mining is useful and important research area in banking sectors for analyzing and identifying important, useful and unknown data of banks [11].

Classification is one of the most important techniques to analyze the raw data, to extract the model, to classify the data into the required forms of outcomes using classification methods, and to predict the categorical labels of data using the prediction models. Support Vector Machine (SVM), is a supervised machine learning model with related learning algorithms analyzing the data, which is used for classification and prediction of data.

In this experiment, the system will classify the bank marketing data using SVM and predict the customers' subscription to the term deposit whether subscribe or not. By using the proposed system, bank can achieve their organizational objectives to increase the number of subscriptions to term deposit.

The outline of this paper is described as follows. The literature reviews of classification on bank marketing data are presented in Section 2. The related background theory of the system is described in Section 3. The system design and explanation of the proposed system is described in Section 4. Performance Evaluation is explained in Section 5. In Section 6, the detail explanations of experimental results are presented. The conclusion of proposed system is presented in Section 7.

2. Literature Reviews

In this section, the various works and literature reviews of classification on bank marketing area are presented.

In the first study [2], Karim Amzile and Rajaa Amzile proposed Support Vector Machine for Smart Direct Marketing to predict bank

customers interested in the Term Deposits in 2021. In this paper, Support Vector Machine was used to predict the behaviors of bank customers toward the term deposits. The dataset was collected during a direct marketing campaign. This model was obtained to minimize the cost and expense of a marketing campaign for banks.

In 2022, Mehmet Furkan Akça and Onur Sevli b [3] used Support Vector Machine to predict the acceptance of the bank loan offers. The authors used to predict results with four kernels of SVM such as Linear, Polynomial, RBF and Sigmoid kernel. The best result was obtained with Polynomial kernel and the lowest success rate was Sigmoid kernel.

The title "A comparative study between support vector machine and support vector data description in bank telemarketing" [4] was presented by Han Gao*; Pei Shan Fam; Heng Chin Low in 2021. The supervised machine learning, SVM and unsupervised machine learning, SVDD were used to compare the prediction performance of bank telemarketing area. The result indicated that the machine learning was more suitable for binary classification problem with the accuracy value 0.9867. Therefore, the authors proposed that SVM model is more suitable for binary classification problem compared to SVDD.

3. Background Theory

The related background theory of this study is described in this section.

3.1. Bank Marketing Data

Bank marketing data refers to the information collected by banks to gain insights into their customers' behavior and preferences, as well as to develop effective marketing strategies. This data includes a variety of information, such as customer demographics, transaction history, account balances, credit scores, and other financial information. By analyzing this data, banks can gain a deeper understanding of their customers and tailor marketing campaigns to specific segments of customers.

One of the uses of bank marketing data is to identify potential customers for various financial products, such as deposits, loans, credit cards, and savings accounts. For example, a bank may use data analytics to identify customers who have a high credit score and a history of responsible financial behavior as potential candidates for a deposit or loan. By targeting these customers with personalized marketing messages, banks can increase the likelihood that they will sign up for these products in [10].

Bank marketing data is a valuable resource that can help banks to understand their customers, develop effective marketing strategies, and improve customer engagement and loyalty. By analyzing this data, banks can identify potential customers for various financial products, personalize marketing messages, optimize marketing budgets, and improve the overall customer experience. However, it's important for banks to handle this data ethically and comply with privacy regulations to protect their customers' personal and financial information.

3.2. Data Pre-processing

In Data Mining, data pre-processing is the most important processing step before going into further process. Data pre-processing is the process of preparing raw data for analysis. It involves handling, cleaning and transforming data to ensure that it is accurate, complete, and suitable for analysis. The quality of the data used for analysis can significantly affect the accuracy and validity of the results obtained.

3.2.1. Handle Missing Values and Remove Duplicate Values

Missing values and duplicate values can badly affect the prediction results. Missing data can be anything such as missing sequence, incomplete feature, files missing, information incomplete, data entry error, etc. Data is cleaned to remove errors, inconsistencies, missing values, duplicates values.

Missing values can be handled by deleting, imputing, or using machine learning algorithms. Duplicate values can be handled by deleting, merging, or identifying and correcting the root cause of the duplication. Therefore, firstly the missing values and duplicate values are needed to handle before going into further process. In this system, X.isnull().sum()*100/len(X) method is used to handle the missing values and to check the duplicate values, str(new_def.duplicated().sum()) method is used.

3.2.2. Handle Outliers

An outlier is a data point that is far away from other related points in the dataset. They can be due to variability in the measurement or can show the experimental errors. Handling outliers is essential because they can significantly affect the analysis and interpretation of the data, leading to inaccurate conclusions and decisions. Outliers can significantly affect the accuracy and validity of the results obtained. Therefore, the outliners are needed to handle and remove them before processing the next steps. Several approaches can be used to handle outliers, including removing them from the dataset, transforming the data, or using robust statistical methods [12]. Box Plot method is used to handle the outliers in this system.

3.2.3. Data Transformation

Data transformation is a fundamental step in data mining which involves the process of converting, cleansing, and structuring data into a usable format that can be analyzed to support decision making processes, and to propel the growth of an organization. Data transformation is used when data needs to be converted to match that of the destination system. It is a crucial aspect of data preparation as it helps to improve data quality, consistency, and relevance for mining purposes. In this system, Label Encoding is used to transform data to the usable format.

3.3. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used statistical technique for reducing the dimensionality of a dataset while retaining most of the variability in the data. It involves finding a linear transformation that maps the original dataset onto a new set of orthogonal variables called principal components. The first principal component explains the maximum amount of variability in the data, and each subsequent component explains as much of the remaining variability as possible. The number of principal components chosen depends on the level of variance that needs to be preserved in [7].

PCA has several applications in data analysis, such as feature extraction, data compression, and visualization. It can help identify patterns and relationships within the data and can be used to improve the performance of machine learning algorithms by reducing the number of input features.

3.4. Classification

Classification is a machine learning technique used to categorize data into predefined classes or categories based on their features or characteristics. A classification algorithm is to solve the inputs when the output is a number of categories or classes based on the labeled data. Classification algorithm can be used when the output is the categorical features such as the customer feedback is 'Positive' or 'Negative' or 'Neutral', the email is 'Spam' or 'Non-spam' and etc. in [5]. There are seven different techniques for classification such as Support Vector Machine, Naïve Bayes, Random Forest, K-Nearest Neighbors, Neural Network, Decision Tree, Regression Analysis. This system uses SVM approach.

3.4.1. Support Vector Machine (SVM)

In Support Vector Machine, each data item is represented as a point in n-dimensional space. The purpose of a support vector classifier is to define an optimal hyperplane to separate the two classes. The hyperplane separates the two classes to determine a plane with the largest margin. SVM is the algorithm that finds a special type of linear model called the maximum margin hyperplane, which gives the maximum separation between decision classes [6].

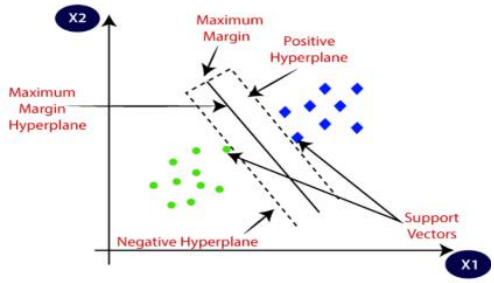


Figure 1. Support Vector Machine

In SVM model, the algorithm tries to find the optimal hyperplane by maximizing the margin, which is the distance between the hyperplane and the closest data points of each class (called support vectors). The algorithm can also use a kernel functions to transform the input data into a higher-dimensional feature space, where a linear hyperplane can separate the data points more effectively.

In SVM, three main lines can be found as follow:

$w^Tx + b = 0$ – for Decision boundary
$w^Tx + b = (-1)$ – for Class-1 boundary
$w^Tx + b = (+1)$ – for Class-2 boundary

1. Positive samples - $w^Tx + b \geq (+1)$
2. Negative samples - $w^Tx + b \leq (-1)$

Then by inducing the variable Y as following, a conditional statement can be generated as follow:

$y = (+1)$ and $y = (-1)$

$$y(w^Tx + b) \geq (+1) \quad (1)$$

Therefore, the support vector is

$$y_i(w^Tx_i + b) = 1 \quad (2)$$

When this condition is satisfied, all the positive and negative data points will be behind the boundary lines. When applying SVM, the maximum width between the boundaries are needed.

$$\text{Width} = (x_{i+} - xi_-)\frac{w^T}{\|w\|} = \frac{(1-b) - (b-1)}{\|w\|} = \frac{2}{\|w\|} \quad (3)$$

Where, $y_i = 1$ is for positive examples and $y_i = -1$ is negative examples.

To find a hyperplane with the maximum margin, which can be expressed as an optimization problem shown as:

$$\text{Minimize: } \frac{1}{2}\|w\|^2 \quad (4)$$

Subject to: $y_i(w^Tx + b) \geq 1$, i = 1,2,…,n

3.4.2. Advantages of SVM

SVM is a powerful machine learning algorithm with several advantages. They are as follows:

- Can handle both linear and nonlinear data.
- Work well with high-dimensional data, and have a strong theoretical foundation.
- SVM has good generalization performance.
- Effectively handle classification and regression tasks.

3.4.3. Disadvantages of SVM

Some of the disadvantages of Support Vector Machine (SVM) includes as follow:

- Sensitive to the choice of kernel function.
- High computational requirements for large datasets.
- Difficult in interpreting the model's results.
- SVM doesn't handle noisy data well.

3.5. Dataset Description

In this system, the bank marketing dataset is used to classify the term 'deposit' of banking industry. The dataset is extracted from the website [8], Kaggle. The dataset contains 11162 rows, 17 columns and two classes (Yes or No). "Age", "Balance", "Day", "Duration", "Pdays", "Campaign" and "Previous" are numeric values and the remainders are categorical values. Using the train-split technique, 80 % of the data will be training data and the remainders are testing data.

4. Proposed System Design

The main goal of the proposed system is to predict the bank customers whether subscribes the term 'deposit' or not by using data mining and its machine learning classification techniques. The overview design of the proposed system is shown in Figure 2. In the system, data pre-processing, feature engineering, splitting dataset as training data and testing data, classification, model building, model evaluation, are performed. The data pre-processing steps will be processed the data handling and transforming to the usable format.

This experiment uses the Support Vector Machine to classify the bank marketing data to predict the customers' subscription of the term 'deposit'. PCA is used for feature engineering. Performance Evaluation is carried out by using Confusion Matrix. The three experimental results: without feature engineering, with feature engineering before using PCA and with feature engineering after using PCA are carried out.

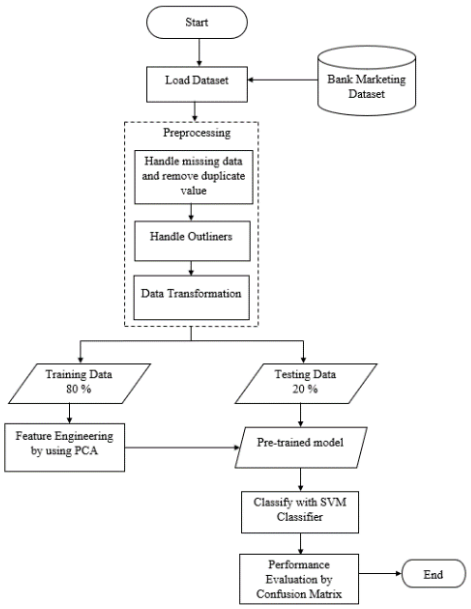


Figure 2. System Flow Diagram

5. Performance Evaluation

In this study, to evaluate the performance of SVM algorithm in classification of bank marketing data, accuracy, precision, recall and F-measure are calculated by using Confusion Matrix.

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

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{Precision + Recall} \quad (8)$$

Where,

TP = correctly predicted deposit
TN = correctly predicted non-deposit
FP = wrongly predicted as deposit
FN = wrongly predicted as non-deposit

6. Experimental Results

The bank marketing dataset undergoes the experiment to determine the effectiveness of the supervised machine learning model for predicting the deposit customer of bank. The proposed system is constructed by using the SVM Classifier. The dataset used in this experiment are 80% for training data and 20% for testing data. According to the results of accuracy, the initial model without feature engineering, the model with feature engineering before using PCA and the proposed model with feature engineering after using PCA are carried out and the proposed model shows the best result.

6.1. Experimental Result without Feature Engineering

The following result is obtained from the initial dataset without feature engineering. The result is shown in Figure 3.

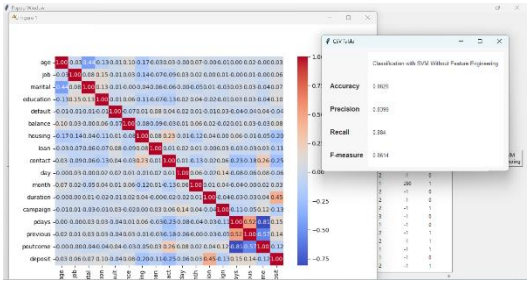


Figure 3. The Result without Feature Engineering Model

The confusion matrix for the model without feature engineering is shown in the following Figure 4.

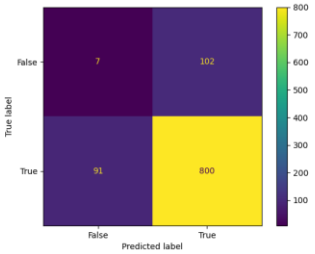


Figure 4. Confusion Matrix without Feature Engineering

6.2. Experimental Result with Feature Engineering Before Using PCA

In Figure 3, the correlation matrix shows that "Previous" is strongly positive correlated with "Pdays". Moreover, "Poutcome" is strongly negative correlated with "Pdays" and "Previous". The 3 features are highly dependent on each other. According to the value of correlations, Pdays is the highest correlation with the target variable, Deposit. The other variables are weak relationship with the target. According to these values, Pdays is kept and the other 2 variables, "Previous" and Poutcome, are dropped from the dataset. The result of the accuracy of the system with feature engineering before using Principal Component Analysis is shown in Figure 5.

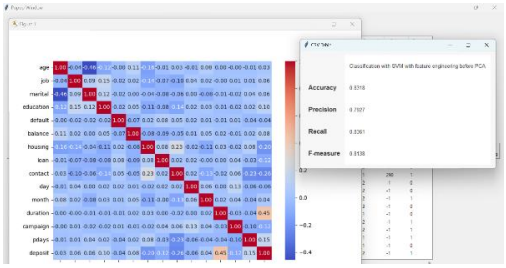


Figure 5. The Result of the System before using PCA

The confusion matrix for the system with feature engineering before using PCA is shown in the following Figure 6.

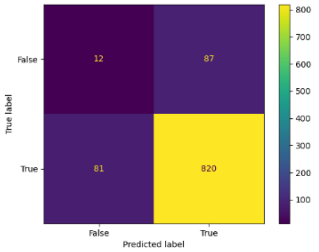


Figure 6. Confusion Matrix with Feature Engineering before using PCA

6.3. Experimental Result with Feature Engineering After using PCA

After dropping the 2 features as seen in Figure 5, "Contact" and "Housing" are positively correlated with the value of 0.23. Therefore, these 2 variables are selected for PCA. In Figure 7, PC1 (the first Principal Component) has the better relationship with deposit than "Contact" and "Housing". Therefore, these 2 variables are replaced with PC1. The result of the accuracy of the system with feature engineering after using PCA is displayed in the following figure.

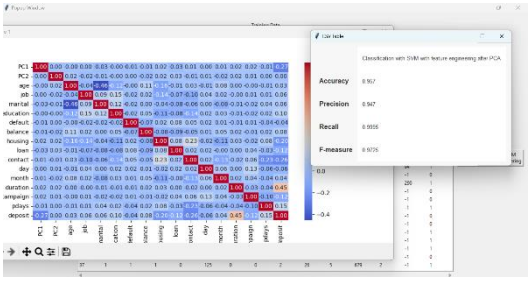


Figure 7. The Result of the Proposed System after using PCA

The confusion matrix for the system with feature engineering after using PCA is shown in the following Figure 8.

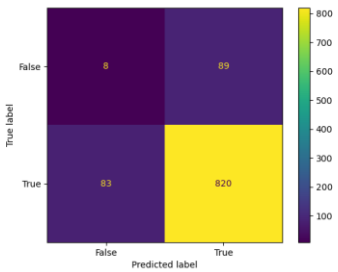
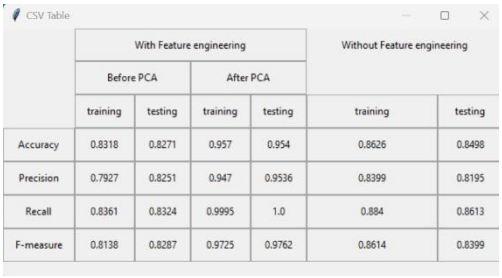


Figure 8. Confusion Matrix with Feature Engineering after using PCA

The final results, the accuracy of the proposed system without feature engineering, the accuracy of the system with feature engineering before

using Principal Component Analysis and the accuracy of the system with feature engineering after using Principal Component Analysis, are shown in the following Figure 9.



CSV Table

	With Feature engineering				Without Feature engineering	
	Before PCA		After PCA			
	training	testing	training	testing	training	testing
Accuracy	0.8318	0.8271	0.957	0.954	0.8626	0.8498
Precision	0.7927	0.8251	0.947	0.9536	0.8399	0.8195
Recall	0.8361	0.8324	0.9995	1.0	0.884	0.8613
F-measure	0.8138	0.8287	0.9725	0.9762	0.8614	0.8399

Figure 9. The Comparison Result of Three Experiments

6.4. Model Comparison

The performance of the initial model, the initial model with feature engineering and the final model are compared and contrasted as shown in the following Figure 10. The final model with feature engineering after using PCA shows the best result in both training data and testing data.

Comparison	Model without Feature Engineering		Model with Feature Engineering Before PCA		Model with Feature Engineering After PCA	
	Training Data	Testing Data	Training Data	Testing Data	Training Data	Testing Data
Accuracy	0.8626	0.8498	0.8318	0.8271	0.9570	0.9540
Precision	0.8399	0.8195	0.7927	0.8251	0.9470	0.9536
Recall	0.8840	0.8613	0.8361	0.8324	0.9995	1.0
F-measure	0.8614	0.8399	0.8138	0.8287	0.9725	0.9762

Figure 10. Models Comparison

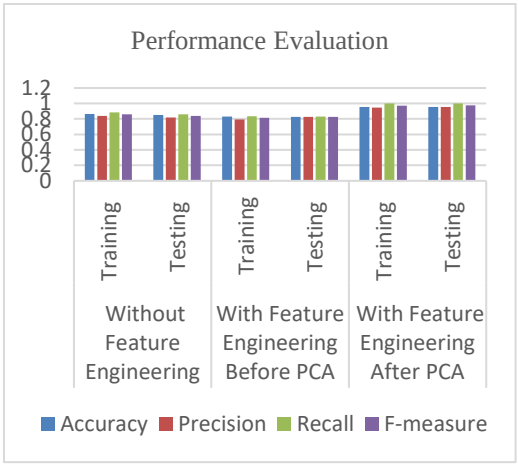


Figure 11. Performance Evaluation of the System

7. Conclusion

In this system, the main objective is to classify the data of the banking industry using Support Vector Machine (SVM) classifier. The accuracy, precision, recall and F-measure are evaluated to compare the system without feature engineering and with feature engineering. The system is implemented by using Python programming language. The analysis of the performance of Support Vector Machine (SVM) is described. The bank marketing dataset is extracted from Kaggle. In this system, the three experimental results: without feature engineering, with feature engineering before using PCA and with feature engineering after using PCA are carried out.

In the first experiment when the training data is used, the accuracy without feature engineering is 86%, the accuracy with feature engineering before using PCA is 83% and the accuracy with feature engineering of after using PCA gets 96%. In second experiment which is used the testing data, the accuracy without feature engineering gets 85%, the accuracy with feature engineering before using PCA is 83% and 95% after using PCA. Therefore, the proposed system (with feature engineering after using PCA) shows the best result of the experiments by comparing and contrasting the accuracies of the results.

References

[1] Jamiu Olalekan Oni, "Exploratory analysis of bank marketing campaign using machine learning; logistic regression, support vector machine and k-nearest neighbor", 2020

[2] Karim Amzile, Rajaa Amzile, "Using SVM for Smart Direct Marketing (SDM): A case of predicting bank customers interested in the Term Deposits", 2021

[3] Mehmet Furkan Akça, Onur Sevli b, "Predicting acceptance of the bank loan offers by using support vector machines", 2022

[4] Han Gao*; Pei Shan Fam; Heng Chin Low, "A comparative study between support vector machine and support vector data description in bank telemarketing", 2021

[5] "Javat Point - Classification and Prediction in Data Mining", from https://www.javatpoint.com/classification-and-predication-in-data-mining

[6] Nachev, T. Teodosiev, " Using Support Vector Machines for Direct Marketing Models", 2015

[7] Jolliffe, I. T, "Principal component analysis", (2nd ed.), Springer, 2011

[8] Janio Martinez Bachmann, "Bank Marketing Dataset" Kaggle, from https://www.kaggle.com/datasets/janiobachmann/bank-marketing-dataset

[9] Karim Amzile, Rajaa Amzile, "Using SVM for Smart Direct Marketing (SDM): A case of predicting bank customers interested in the Term Deposits", 2021

[10] Bankinter, "What are Bank Deposits?", from https://www.bankinter.com/banca/en/faqs/investment-products/what-are-bank-deposits

[11] A. Nachev, T. Teodosiev, "Using Support Vector Machines for Direct Marketing Models", April, 2015

[12] Harika Bonthu, "Detecting and Treating Outliers | Treating the odd one out!", April, 14, 2023

The Car Insurance Claim Prediction System by using Machine Learning Algorithms on Apache Spark Platform

Thein Than Ko, Tin Zar Thaw
University of Computer Studies, Yangon
theinthanko.001mgy@gmail.com, tinzarthaw@ucsy.edu.mm

Abstract

The car insurance companies face a major challenge in dealing with insurance claims, which are prone to fraud and increasing in volume. This makes it difficult for insurers to classify claims during the review process. To address this issue, the aim of this study was to develop four Car Insurance Claim Prediction Classifiers using Random Forest and Logistic Regression with and without Variance Threshold Selector Method for identifying important attributes for accurate prediction. The dataset is divided into training and testing sets, and classifiers are created using all attributes. Additionally, classifiers are created using the feature selection method by removing low variance attributes. The system analyzes different attribute numbers and evaluates the classifiers based on metrics such as accuracy and f-score. The results show that Random Forest classifiers, both with and without the feature selection method, are more suitable for the proposed system than Logistic Regression classifiers. Among different attribute numbers, classifiers based on 38 and 40 attributes perform the best, while the classifier based on 42 attributes is the second best.

Keywords: Logistic Regression, Random Forest, Variance Threshold Selector Method, Apache Spark

1. Introduction

The insurance industry is a fast-growing sector that plays a crucial role in ensuring the economic well-being of a country. However, car insurance claims in insurance companies can be costly problems, and insurance providers must always make a great effort to combat the growing cost of insurance claims and claim loss due to insurance claim fraud [7]. Insurance companies face business problems, such as risk assessment, classification of policy holders, resource allocation, insurance claim classification, and prediction in the insurance claim handling process. With the advancements in computing technology, machine learning approaches have emerged as a viable solution to these problems, particularly for handling and processing large amounts of data such as that found in insurance databases [1].

The use of machine learning classifiers in big data analysis helps the insurance industry to predict future trends in the competitive market. Big data, which includes structured, unstructured, and semi-structured data, has fundamentally changed data management across the insurance industry [8] as traditional relational database management systems and software tools are unable to cope with the sheer volume and variety of data [2]. In this system, Apache Spark, open source processing engine, uses to control big data problem. Spark helps in some challenging and computationally exhaustive tasks like processing high volumes of real-time and archived data, thereby integrating the complex capabilities such as ML and graph algorithms. In this system, Logistic Regression Classifier, Random Forest Classifier and Variance Threshold Selector method from MLib are used to apply the car insurance claim prediction system.

This paper is organized as follows: Section 2 discusses related works and Section 3 explains background theory. Data Collection and Preprocessing of Car Insurance Claim Dataset are discussed in Section 4 and Implementation of the proposed system and experimental result are described in Section 5 and Section 6. Finally, Section 7 presents conclusion and Section 8 presents advantages and future works.

2. Related Works

The system utilizing Random Forest (RF) and Multi Class - Support Vector Machine (SVM) was developed for Motor Insurance Claim Status Prediction [5]. The dataset used consists of eleven attributes related to motor insurance claim data from AIC Company, with five target classes: close, notification, pending, re-open, and settled. In the domain of insurance, particularly motor

insurance, the RF model exhibited slightly superior prediction accuracy compared to the SVM model. The proposed system [12] introduced a hybrid predictive modeling approach for motor insurance claims by combining grey relational analysis with backpropagation neural network (BPNN). The study provided evidence that, considering different numbers of features and hidden nodes to rank informative features, GRABPNN outperformed other models in modeling claim frequency and claim severity for each claim type.

The authors describe a proposed system for predicting the occurrence of motor insurance claims [4]. They treated this prediction as an imbalanced machine learning problem and utilize various algorithms such as Logistic Regression, Decision Tree, Random Forest, XGBoost, and Feed-forward Network. The authors mentioned that even high-performing machine learning algorithms may not provide satisfactory results with imbalanced data. Among the algorithms tested, XGBoost and Random Forest methods demonstrate superior accuracies compared to the others. In the proposed system [9], a prediction model for auto insurance claims was developed using various machine learning techniques, including Artificial Neural Network (ANN), Decision Tree (DT), Naïve Bayes classifiers, and XGBoost. Notably, the XGBoost model and Resolution Tree exhibited the highest accuracy among the four models, achieving an accuracy of 92.53% and 92.22%, respectively.

In this paper, the car insurance claim prediction models using Logistic Regression, Random Forest and Variance Threshold Selector method are presented.

3. Background Theory

Apache Spark is an open-source processing engine. Spark utilizes a directed acyclic graph and its own data structure called Resilient Distributed Dataset (RDD) for high-speed processing and analytics [14]. Spark integrates machine learning (ML) and graph algorithms, enabling advanced analytics. It includes a dedicated ML library called MLib, which offers algorithms for classification, regression, clustering, collaborative filtering, and dimensionality reduction. In the proposed system, the car insurance claim prediction system is developed using Spark's MLib library. The system incorporates a feature selection method and employs two classification methods.

3.1. Feature Selection with Variance Threshold Selector Method

The system uses Feature selection method, Variance Threshold Selector to choose a subset of the most important car insurance claim features while trying to retain as much information as possible for car insurance companies. This technique is a quick and lightweight way of eliminating features with very low variance, i. e. features with not much useful information [6][10]. Variance Method:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad (1)$$

Where, n is the number of records, xi is the value at position i and x is the mean of particular attribute. The proposed system calculated car insurance claim attributes' variances and removes attributes with low variance according to the user specified needed attributes number.

3.2. Logistic Regression Classifier

Logistic regression is the algorithm for classification. Spark's logistic regression API is useful for binary classification, or classifying input data into one of two groups [14]. The proposed system accepts car insurance claim information as input. To predict car insurance claim, the process of logistic regression produces a logistic function. Use the function to predict the probability that an input vector belongs in one group or the other. The Logistic regression equation can be obtained from the Linear Regression equation:

log[y/(1-y)]=b0+b1x1+b2x2+…….+bnxn (2)

where b are the regression coefficients, xi is the real data and y is the predict class data of the particular record. Finally, this proposed system uses Logistic Regression Classifier to predict the car insurance claim condition.

3.3. Random Forest Classifier

The Random Forest algorithm operates in two phases. Firstly, it creates a random forest by combining N decision trees. Secondly, it utilizes these trees to make predictions for each tree generated in the initial phase. To ensure accurate results, it is crucial that the predictions from each tree exhibit minimal correlations [3][11]. Random Forest offers several advantages, including:

- It requires less training time compared to other algorithms.
- It achieves high prediction accuracy, even when dealing with large datasets, ensuring efficient performance.
- It can handle missing data effectively without compromising accuracy.

In this system, the classifier creates decision trees according to the car insurance claim dataset and then creates the random forest by combining car insurance claim decision trees to predict accurate results.

4. Data Collection and Preprocessing of Car Insurance Claim Dataset

The dataset is collected from the Kaggle website with 44 attributes and 97656 cases [13]. The proposed system makes the data pre-processing step manually and convert their value types of attributes: String and Boolean into number type. In preprocessing step, string type attributes are converted into number data according to equations (3) to (10).

$$f(area\_cluster) = \begin{cases} 1, \ if \ (area\_cluster = "C1") \\ 2, else \ if \ (area\_cluster = "C2") \\ 3, else \ if \ (area\_cluster = "C3") \\ . \\ . \\ 21, else \ if \ (area\_cluster = "C21") \\ 22, \ (otherwise) \end{cases} \quad (3)$$

Where f($area\_cluster$) function is converted the Area_cluster attribute into number data based on x's string value.

$$f(segment\_attribute) = \begin{cases} 0, if \ (segment\_attribute = "A") \\ 1, else \ if \ (segment\_attribute = "B1") \\ 2, else \ if \ (segment\_attribute = "B2") \\ 3, else \ if \ (segment\_attribute = "C1") \\ 4, else \ if \ (segment\_attribute = "C2") \\ 5, (otherwise) \end{cases} \quad (4)$$

Where f($segment\_attribute$) function is converted the $segment\_attribute$ e of the car into number data based on segment_attribute's string value.

$$f(rear\_brakes\_type) = \begin{cases} 0, \ if \ (rear\_brakes\_type = "Drum") \\ 1, \ (otherwise) \end{cases} \quad (5)$$

Where f($rear\_brakes\_type$) function is converted the value's data type of the rear brakes type attribute into number data type based on their string value.

$$f(transmission\_type) = \begin{cases} 0, if \ (transmission\_type = "Manual") \\ 1, else \ if \ (transmission\_type = "Electric") \\ 2, (otherwise) \end{cases} \quad (9)$$

Where f(transmission_type) function is converted the value's data type of the transmission type attribute into number data type based on fuel type attribute's string value.

$$f(model) = \begin{cases} 1, if \ (model = "M1") \\ 2, \ else \ if \ (model = "M2") \\ 3, \ else \ if \ (model = "M3") \\ . \\ . \\ 9, \ else \ if \ (model = "M10") \\ 10, (otherwise) \end{cases} \quad (6)$$

Where f(model) function is converted the value's data type of the car model attribute into number data type based on model's string value.

$$f(engine\_type) = \begin{cases} 0, \quad if \ (engine\_type = "1.0 \ Sce") \\ 1, \quad else \ if \ (engine\_type = "1.2 \ L \ K \ Series \ Engine") \\ 2, \quad else \ if \ (engine\_type = "1.2 \ L \ K12N \ Dualjet") \\ 3, \quad else \ if \ (engine\_type = "1.5 \ L \ U2 \ CRDi") \\ 4, else \ if \ (engine\_type = "1.5 \ Turbocharged \ Revotorq") \\ 5, else \ if \ (engine\_type = "1.5 \ Turbocharged \ Revotron") \\ 6, else \ if \ (engine\_type = "F8D \ Petrol \ Engine") \\ 7, else \ if \ (engine\_type = "G12B") \\ 8, else \ if \ (engine\_type = " \ i - DTEC") \\ 9, else \ if \ (engine\_type = "K \ Series \ Dual \ jet") \\ 10, (otherwise) \end{cases} \quad (7)$$

Where f(engine_type) function is converted the value's data string type of engine type attribute into number data type based on engine type of the car.

$$f(steering\_type) = \begin{cases} 0, if \ (steering\_type = "Electric") \\ 1, else \ if \ (steering\_type = "Power") \\ 2, (otherwise) \end{cases} \quad (10)$$

Where f(steering_type) function is converted the value's data type of the power steering present in the car into number data type based on steering_type attribute's string value.

5. Implementation of the Proposed System

The Figure 1 depicts an overview of the proposed system. The system predicts car insurance claim of customers by using LR and RF classifiers with or without using a feature selection method. To predict car insurance claim with the feature selection method, the first task is that the system takes the car insurance claim dataset and selects the important features by using Variance Threshold Selector method based on user specified feature number. Secondly, the system splits the selected features data set into two datasets: training dataset with 80% and testing dataset with 20% and then creates two classifiers: Linear Regression Classifier and Random Forest Classifier based on training dataset respectively.

Finally, the system provides the particular prediction accuracy of two classifiers based on testing dataset. To predict car insurance claim without the feature selection method, the system takes the car insurance claim dataset and splits two data sets: training dataset with 80% and testing dataset with 20%. After preparing dataset, the system creates Linear Regression Classifier and Random Forest Classifiers and predict the accuracy of two classifiers based on testing dataset.

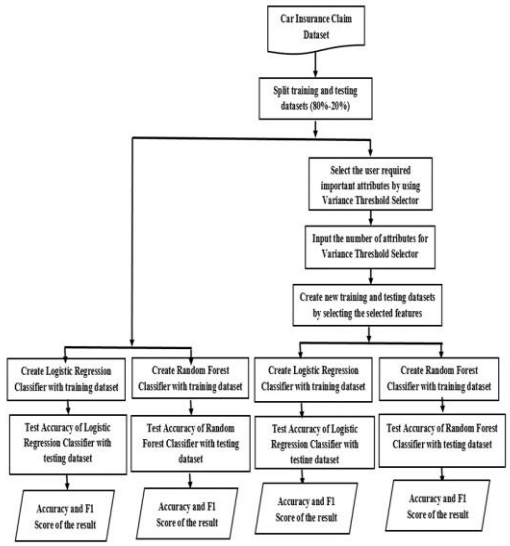


Figure 1. Proposed System Design of the Car Insurance Claim Prediction System based on Accuracy Measuring

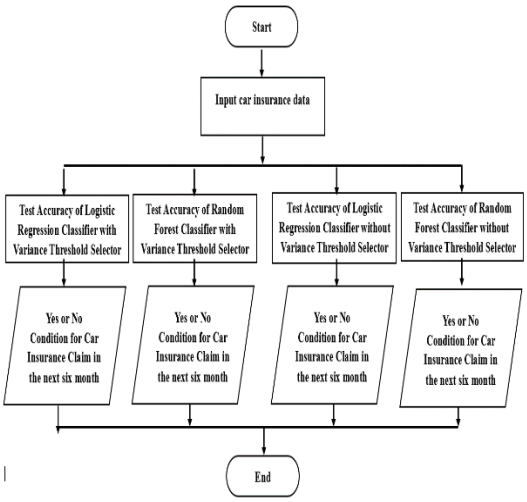


Figure 2 Proposed System Design of the Car Insurance Claim Prediction System based on User Input Car Insurance Data

Proposed System Design of the Car Insurance Claim Prediction System based on User Input Car Insurance Data is shown in The Figure 2. To predict car insurance claim or not in the next six months, the user inputs the car insurance claim data and predict yes or no results by using Logistic Regression Classifier with attribute selection, Random Forest Classifier without attribute selection, Logistic Regression Classifier without attribute selection and Random Forest Classifier with attribute selection.

6. Experimental Results of the System

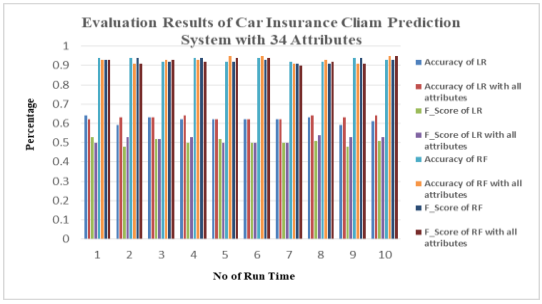


Figure 3. Comparison of 34 Attributes Evaluation Results Based on 10 Tests Dataset

To analyze the experiment result, the number of attributes: 34, 36, 38, 40, 42 are taken to measure the accuracy and f-score and each attribute selection run ten times because of splitting training and testing randomly. For the number of 34 attributes based on 10 tests, accuracy and f-score results of LR classifier are less than 64 percent and accuracy and f-score results of RF classifier are greater than 86 percent that are shown in Figure 3. Moreover, LR classifiers of other attributes numbers are not suitable for this car insurance claim dataset nature because of their accuracy and f-score values with lower than 64 percent. RF classifier is suitable for this proposed system.

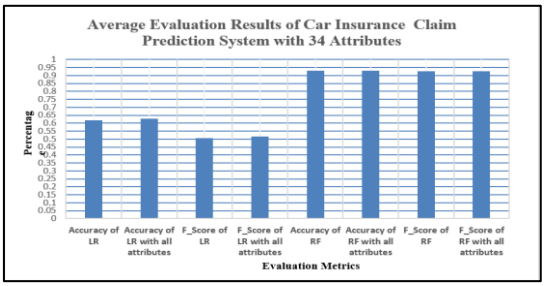


Figure 4. Comparison of Accuracy and F-Score Values for LR and RF Based on 34 Attributes

Figure 4 accuracy and f-score results of LR classifiers are below 63 percent. On the other hand, the accuracies of RF with attribute selection and without attribute selection are 0.931 percent and 0.93 percent respectively. The f-scores of RF with attribute selection and without attribute selection are 0.927 percent and 0.925 percent respectively. For 34 attributes, evaluation metric values of two

classifiers: RF with and without attributes selection method are produce similar results.

In Figure 5, the LR classifiers achieve accuracy and f-score results below 63 percent. The accuracies of RF classifiers with attribute selection and without attribute selection are 0.952 percent and 0.947 percent respectively. Additionally, the f-scores of RF classifiers with attribute selection and without attribute selection are 0.949 percent and 0.941 percent respectively. Consequently, the RF classifier with the attribute selection method is deemed more suitable for the proposed car insurance claim prediction system, which is based on 36 attributes.

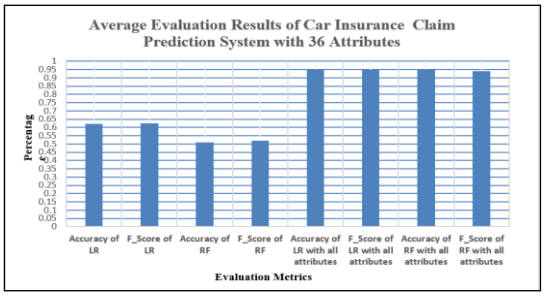


Figure 5. Comparison of Accuracy and F-Score Values for LR and RF Based on 36 Attributes

In Figure 6, accuracy and f-score results of LR classifiers are less than 63 percent. Accuracies of RF with the attribute selection method and without the attribute selection method are 0.965 percent and 0.932 percent respectively. F-scores of RF with the attribute selection method and without the attribute selection method are 0.965 percent and 0.927 percent respectively.

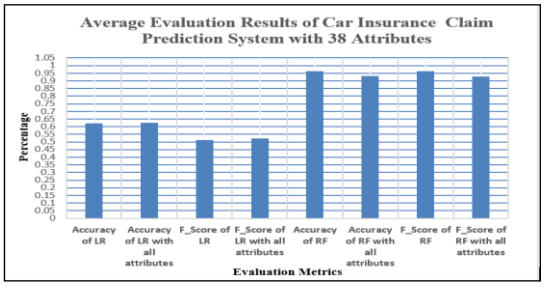


Figure 6. Comparison of Accuracy and F-Score Values for LR and RF Based on 38 Attributes

Figure 7 shows accuracy and f-score results of LR classifiers that are less than 62 percent. Accuracies of RF with the attribute selection method and without the attribute selection method are 0.966 percent and 0.948 percent respectively. F-scores of RF with the attribute selection method and without the attribute selection method are 0.965 percent and 0.945 percent respectively. Therefore RF classifier with the attribute selection method is more suitable for the proposed car insurance claim prediction system based on both 58 and 40 attributes.

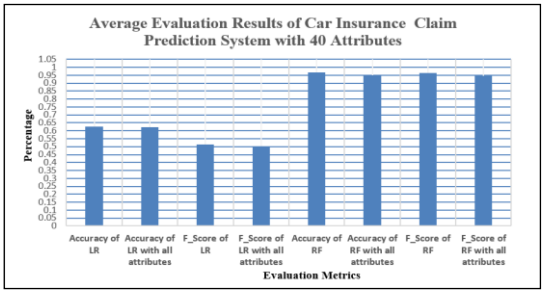


Figure 7. Comparison of Accuracy and F-Score Values for LR and RF Based on 40 Attributes

Figure 8 demonstrates accuracy and f-score results of LR classifiers are less than 63 percent. Accuracies of RF with the attribute selection method and without the attribute selection method are 0.952 percent and 0.939 percent respectively. F-scores of RF with the attribute selection method and without the attribute selection method are 0.946 percent and 0.935 percent respectively. Therefore RF classifier with the attribute selection method is more suitable for the proposed car insurance claim prediction system based on 42 attributes.

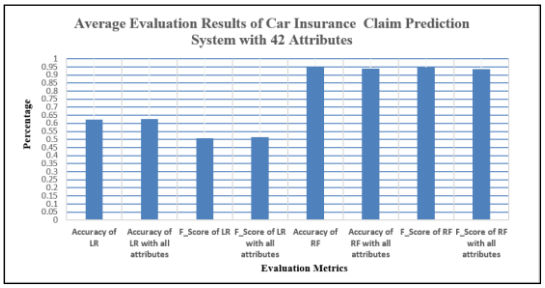


Figure 8. Comparison of Accuracy and F-Score Values for LR and RF Based on 42 Attributes

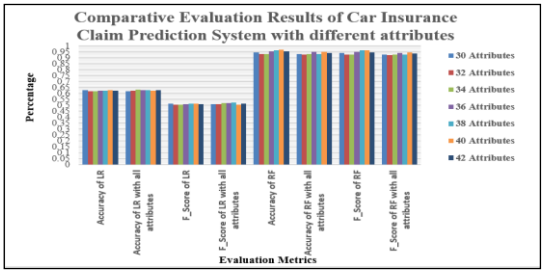


Figure 9. Comparison of Accuracy and F-Score Values for LR and RF Based on 42 Attributes

According to the overall experimental results in Figure 9 from comparing accuracy and f-score values of car insurance claim prediction system

with different attributes, the classifiers based on 38 attributes and 40 attributes are the best classifiers with the same accuracy (0.97 percent) and the same f-score (0.97 percent) that are shown in Figure 4.15. The second best classifier based on 42 attributes are accuracy with 0.95 percent and 0.95 percent.

7. Conclusion

This system has proposed four classifiers: Linear Regression (LR) based on Variance Threshold Selector with selected attributes, LR classifier with all attributes, Random Forest (RF) classifier based on Variance Threshold Selector with selected attributes, RF classifier with all attributes. To create four classifiers, the system has divided the dataset into training dataset with 80% and testing dataset with 20% randomly. For the two classifiers with all attributes, the training dataset is used to create the LR classifier and RF classifier. For two classifier with the feature selection method, the new training dataset and testing dataset by removing low variance value of attributes using Variance Threshold Selector method. After that, two LR classifier and RF classifier are been created by using new datasets. The system has analyzed the difference attributes: 34, 36, 38, 40 and 42 to choose the number of attributes and important attributes. The system has tested 10 times for each attribute number because of splitting training and testing datasets randomly. Finally the system compares the evaluation results with metrics: accuracy and f- score. LR classifiers with and without the feature selection method are not suitable for the car insurance claim prediction system because the dataset is not linearly separable the two classes of data from each other and their accuracies are not greater than 0.65 percent. For RF classifiers with and without the feature selection method are suitable for the proposed system with accuracy 93 percent and f-score 92 percent. Moreover, the classifiers based on 38 attributes and 40 attributes are the best classifiers with the same accuracy (0.97 percent) and the same f-score (0.97 percent) while the second best classifier based on 42 attributes are accuracy with 0.95 percent and 0.95 percent.

8. Limitation of the system and Future Work

The system has the limitations of using Logistic Regression as a classifier due to the nature of the dataset, with accuracy results below 63%. The system suggests the need to explore alternative approaches that are suitable for the dataset and mentions the challenges associated with choosing input attributes. Moreover, the performance of this system can be evaluated by using various evaluation methods. In the future, this system can be used to test a large amount of training data and testing data.

References

[1] A. C. Yeo, K. A. Smith, R. J. Willis, and M. Brooks, “Clustering Technique for Risk Classification and Prediction of Claim Costs in the Automobile Insurance Industry,” Int. J. Intell. Syst. Accounting, Financ. Manag., no. November 1999, pp. 39–50, 2001.

[2] A. L. Heureux and M. Grolinger, Katarina and Caprtz, “Machine Learning With Big Data : Challenges and Approaches,” IEEE Access, vol. 5, pp. 7776–7797, 2017.

[3] A. S. Alshamsi and A. Ain, “Predicting Car Insurance Policies Using Random Forest,” IEE, pp. 128–132, 2014.

[4] B. Sebastian, and P. Rola, "Prediction of motor insurance claims occurrence as an imbalanced machine learning problem.", Cornell University, arXiv preprint arXiv:2204. 06109 (2022).

[5] E. Alamir, T. Urgessa, A. Hunegnaw, T. Gopikrishna, “Motor Insurance Claim Status Prediction using Machine Learning Techniques”, International Journal of Advanced Computer Science and Applications, Vol. 12, No. 3, 2021.

[6] E. Alamir, T. Urgessa, T. GopiKrishna and Ellappan V, “Application of Machine Learning with Big Data Analytics in the Insurance,” vol. 11, no. 12, pp. 1064–1073, 2020.

[7] M. C. Wijegunasekara and Weerasingheand, “A Comparative Study of Data Mining Algorithms in the Prediction of Auto Insurance Claims,” vol. 5, no. 1, pp. 47–54, 2016.

[8] P. Bharal and A. Halfon, “Making Sense of Big Data in Insurance,” ACORD and MarkLogic, 2013.

[9] S. Abdelhadi, K. Elbahnasy, M. Abdelsalam, “A Pproposed Model To Predict Auto Insurance Claims Using Machine Learning Techniques”, Journal of Theoretical and Applied Information Technology 30th November 2020. Vol.98. No 22.

[10] T. Kavipriya and N. Kumar, “A Study on Machine Learning Algorithms for Big Data Analytics,” IOSR J. Eng., no. Iccids, pp. 40–46, 2018.

[11] W. Lin, Z. Wu, L. Lin, A. Wen, and J. I. N. Li, “An Ensemble Random Forest Algorithm for Insurance Big Data Analysis,” IEEE Acess, vol. 5, 2017.

[12] Z. M. Yunos, S. M. Shamsuddin, R. Sallehuddin, and R. Alwee, “Hybrid Predictive Modelling for Motor Insurance Claim”, Joint Conference on Green Engineering Technology & Applied Computing”, 2019.

[13]Car Insurance Claim Prediction Dataset from Kaggle:https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification.

[14]“Documentation | Apache Spark”, https://spark.apache.org/documentation.html.

Secure Video Steganography by Using Knight Tour Algorithm and Least Significant Bit (LSB) Method

Aye Thandar Htun, Nwe Thazin
University of Computer Studies (Taunggyi)
ayethandarhtun@ucstgi.edu.mm, nwethazin@ucstgi.edu.mm

Abstract

Steganography is the practice of hiding secret data within other types of digital media, such as images or audio, to ensure secrecy between senders and receivers. The main goal is to add additional data to digital content without it being noticeable. This paper proposed an enhanced video steganographic system that combines cryptography, LSB, and the Knight tour algorithm to improve the security and imperceptibility of hidden data. The system uses cryptography and the Knight tour algorithm to randomize the order of pixel selection, making it more difficult to detect hidden data. The proposed method has strong security and satisfactory values of PSNR, MSE, and SSIM, compared to the basic LSB and AES-LSB methods.

1. Introduction

Steganography is the practice of hiding secret information in other types of digital media, such as images or audio, to ensure secrecy between senders and recipients. It is a more secure way to transmit data than encryption alone, as it can hide the presence of the data altogether [1]. It is important to note, however, that steganography is not foolproof. With enough time and effort, a skilled steganalyst can often detect hidden messages. Therefore, steganography should not be used as the only way to protect information. It should be used in conjunction with other security measures, such as cryptography.

Traditionally, steganography was used to hide secret information in images. However, recent research has shown that it can also be used to hide data in video files. This is because video files are more complex than images, making them more difficult to analyze for hidden data. One of the most common methods of steganography is the least significant bit (LSB) method. This method hides the secret data by replacing the least significant bits of the cover media with the bits of the secret message. The LSB method is simple and straightforward, but it is not very secure. The main disadvantages of this technique are the serial selection of pixels within the frame that is used for embedding the information inside it.

The proposed system combines video steganography and encryption techniques to create a highly secure system for confidential data. The combination of encryption and steganography can provide a very high level of security for data hiding. In this system, a system that combines AES encryption and LSB technique with Knight tour algorithm is considered to address the limitations of LSB steganography.

2. Related Works

The existing literature offers valuable insights into various data hiding and steganography techniques in digital media.

In [2], a lossless data hiding technique relying on LSB manipulation of digital images was proposed. The LSB algorithm was used in the spatial domain to subtly embed secret data within the least significant bits of the cover image, preserving image visibility. M. Ramalingam's work in [3] introduced a steganographic method to hide text in video files by modifying the least significant bits (LSBs). This method is imperceptible to the human eye and can be used to hide confidential information.

Modified LSB-based steganography improves the security of LSB-based steganography by using different techniques, such as bit plane slicing and hash-based LSB insertion. For instance, [4] introduced a modified LSB algorithm that segments the cover image into byte groups and leverages mixture distributions for embedding secret image bytes. The embedded bits were distributed through the odd bytes of the

image, reducing visibility. Another variant, proposed by A. Singh and H. Singh in [5], divides RGB images into parts and employs bit plane slicing to embed secret messages in a 2:2:4 ratio across the red, green, and blue segments. This approach minimizes noise and perceptible changes. On the Android platform, [6] presented a video steganography scheme utilizing Pixel Differencing Value (PDV) and LSB substitution methods to securely insert secret messages into videos.

Other approaches have also been proposed that combine steganography with other techniques, such as cryptography and compression. Within the framework of [7], authors proposed an image-hiding technique in videos that modifies one, two, or three LSBs per pixel in video frames. The added layer of AES encryption further bolsters the image's protection. Similarly, [8], P. K. Shafna introduced an innovative method that uses AES encryption to secure text data and pixel value differencing for text embedding within video frames. This technique optimizes remainder alteration and circumvents boundary issues, resulting in improved PSNR and MSE outcomes.

3. System Architecture

The proposed system combines AES-128 bits encryption, LSB steganography, and the Knight tour algorithm to create a highly secure system for confidential data.

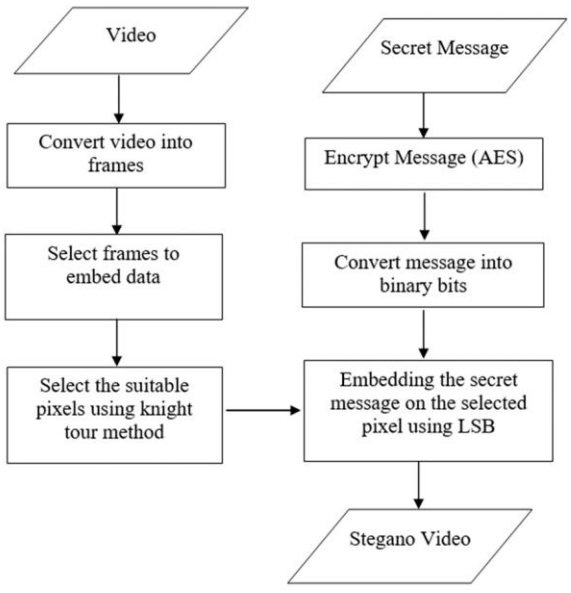


Figure 1. System Architecture

The idea is that the standard AES encryption will make it more difficult for unauthorized users to access the data, and the LSB steganographic technique will make it difficult for unauthorized users to detect the presence of the data. Then, the Knight tour algorithm can be used to randomly select pixels in the carrier medium, which will make it more difficult for attackers to detect or extract the hidden data. This approach to data hiding can be used to hide data in a variety of multimedia files, such as images, videos, and audio files. The general architecture of the system is shown in figure 1.

In this system, the secret data or message is encrypted using the AES-128 bits encryption algorithm. This ensures that the data is secure and cannot be easily read by unauthorized users. The encrypted data is then embedded into the video frames using a steganography algorithm. The steganography algorithm selects pixels in a random order using the Knight tour algorithm in the video frames and modifies their least significant bits (LSBs) to represent the hidden data. This ensures that the hidden data is spread out evenly across the video frames, making it more difficult for attackers to detect. The steganographic video is then transmitted over a communication system to the receiver.

3.1. Embedding Process

Using the Least Significant Bit (LSB) technique, the encrypted secret message is embedded. The LSB technique, a steganographic method, conceals data within the least significant bits of an image. In this context, the chosen pixels derived from the knight tour algorithm (KTA) will store the encrypted message. The proposed technique divides the cover video into 8x8 pixel blocks. This is done to make the embedding process more efficient. Each pixel block is considered as an 8x8 chessboard. This allows the proposed system to use the knight tour algorithm to select pixels for embedding the hidden data. The knight tour algorithm ensures that the hidden data is spread out evenly across the cover video, making it more difficult for attackers to detect. The embedding processes is iterated until the entire encrypted secret message becomes part of the cover frame. Figure 2 illustrates the flow of embedding process of proposed video steganography system.

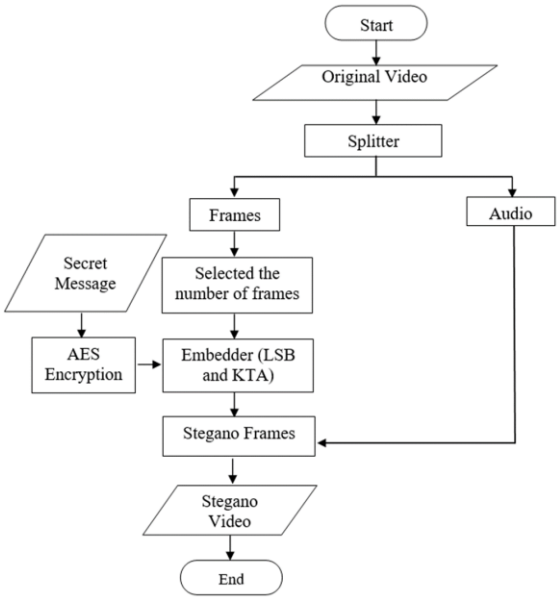


Figure 2. Embedding Process of Proposed Video Steganography System

Upon successful embedding, the images are reconverted into frames. To complete the procedure, the frames are merged to form the steganographic video.

3.2. Extraction Process

After the embedding process is completed, the sender sends the stegano video to the recipient. The recipient divides the stegano video into frames and uses the knight tour algorithm to identify the pixels that contain the hidden data.

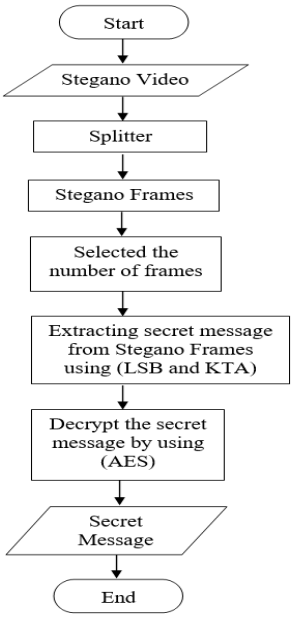


Figure 3. Extraction Process of Proposed Video Steganography System

The LSB method is then used to extract the encoded secret message from the LSBs of the video frames. The extracted message is decrypted by applying the AES algorithm to the message, using the same secret key that was used to encrypt the message. This results in the original secret message. Figure 3 illustrate the flow of extraction process of proposed video steganography system.

4. Evaluation Metrics

The proposed system is evaluated using three metrics: the similarity between the cover and stegano video (SSIM), the picture (distortion) (PSNR) and the mean squared error (MSE). In steganography, there is a trade-off between security and usability. The hidden data must be secure, but the host video must also be visually appealing. To achieve this balance, imperceptibility and visual quality must be seamlessly integrated. This can be done by using metrics such as PSNR, MSE, and SSIM. The equations for these metrics are as follows:

$$\mathrm{MSE} = (\mathrm{m} \times \mathrm{n})^{-1} \sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n}} [\mathrm{I}(\mathrm{i},\mathrm{j}) - \mathrm{K}(\mathrm{i},\mathrm{j})]^2 \quad (1)$$

Where, “m, n” is the dimensions of cover the image and “I” is the original Image and “K” is stegano-image. After calculating “MSE”, its value is used in the calculation of “PSNR”.

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \quad (2)$$

Where the “MAX^2” represents the maximum possible difference in the numerical representation of each pixel.

$$\mathrm{SSIM} = \frac{(2\mu_x\mu_y + (k_1 L)^2)(2\sigma_{xy} + (k_2 L)^2)}{((\mu_x)^2 + (\mu_y)^2 + (k_1 L)^2)((\sigma_x)^2 + (\sigma_y)^2 + (k_2 L)^2)} \quad (3)$$

The default configuration for “SSIM” uses k_1=0.01, k_2= 0.03 and return values in the range [-1.0,1.0] where 1.0 indicates the images are identical. The “SSIM” can be computed using the means “μ_x” and “μ_y”, variances, $(\sigma_x)^2$ and $(\sigma_y)^2$, and covariance "σ_{xy}" of the images.

5. Experimental Results

The performance of the proposed video steganographic system was evaluated by

considering three main factors: imperceptibility, payload, and execution time. A comparative study was conducted with two other methods: a basic LSB method and a combined approach of cryptography and steganography called AES-LSB. All experiments were conducted on a laptop with an Intel Core i5 processor, 4 GB of RAM, and 1 TB HDD. The programming language used throughout the experiments was Python 3 with the OpenCV, NumPy, and FFmpeg libraries. Table 1 presents the characteristics of the different videos that were used.

Table 1. Comparing the Different Video

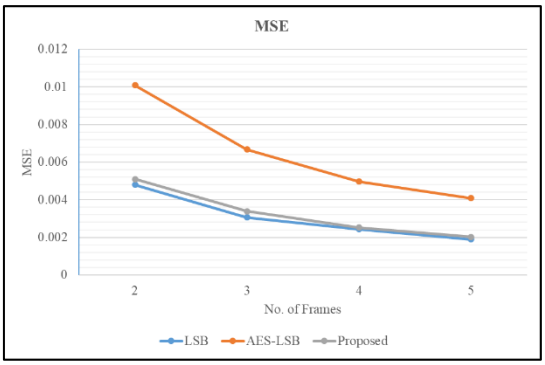
No	Video	Resolution	Duration (sec)	No. of Frames
1	Sunset. mp4	640*360	3	94
2	Small. mp4	560*320	5	168
3	Nature .mp4	640*360	7	170
4	Sunris e.mp4	480*360	9	235
5	Bunny. mp4	320*240	13	206

5.1. Comparison of Imperceptibility

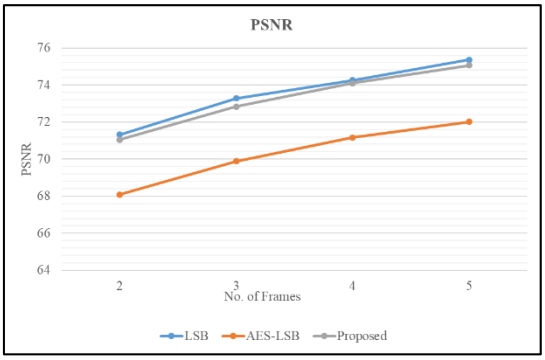
The comparison of imperceptibility is a critical consideration because it ensures both the security of hidden data and the visual appeal of the video. If the hidden data is too noticeable, it may be detected by unauthorized individuals. If the video quality is too degraded, the hidden data may be difficult to recover. By combining imperceptibility and visual quality, a steganographic system can achieve a balance between security and usability.

To evaluate the potential impact of the concealed message on video quality, a comparative analysis was performed between the original video and the steganographic video. As a testing video, “Sunset.mp4” a 640x360 resolution video, was used. The evaluation involved testing the hidden message with different frame numbers, such as 2, 3, 4, and 5 frames. The imperceptibility was measured using the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM).

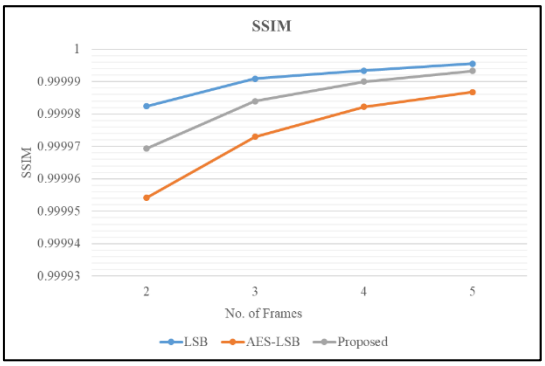
According to the experiment shown in figure 4, the proposed method achieved average image quality preservation among the three methods, even as the number of frames increases. It tended to produce the lowest MSE values across different frame numbers, which means that it may have introduced the least amount of distortion to the cover image. It also produced the second-highest PSNR values, which means that it was able to hide the data without significantly affecting the image quality. The proposed method also produced the highest SSIM values, which means that it was able to preserve the visual fidelity of the image.



(a)



(b)



(c)

Figure 4. Comparison of MSE (a), PSNR (b) and SSIM (c) for the Proposed Method, Simple LSB, and Combined AES-LSB

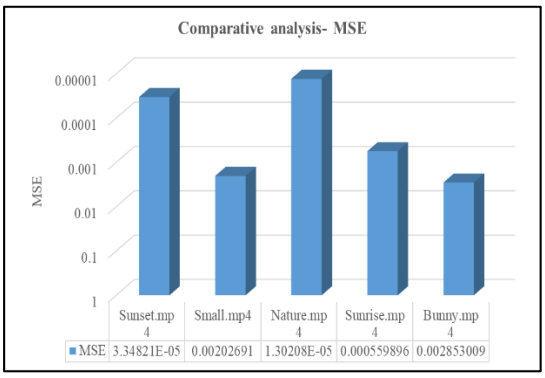
The LSB method also produced good results, but its performance started to plateau as the number of frames increased. The AES-LSB method produced slightly lower results than the

LSB method, but it was still a good option for applications where security is more important than image quality. Overall, the proposed method is the best option for applications where both image quality and security are important.

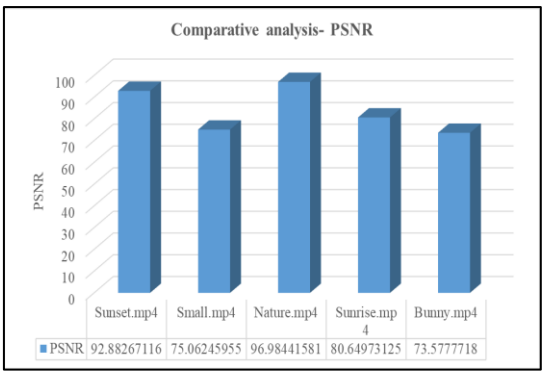
5.2. Comparative Analysis of Video Files

To compare the impact of video file resolution on image quality preservation, an experiment was conducted using 5 videos with different resolutions. Video file resolution is the number of pixels that make up a video. Higher resolution videos have more pixels, which leads to better image quality, but requires more processing power. This can be a challenge for video files, as they are often large and complex.

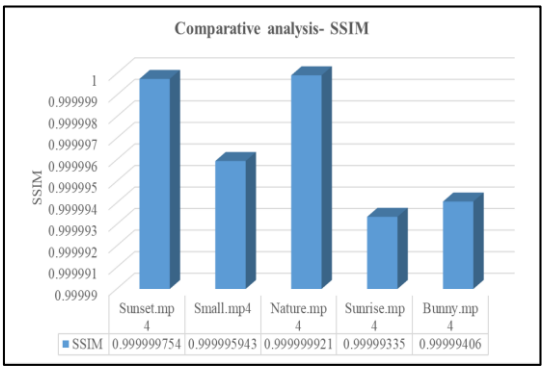
Figure 5 illustrates that the resolution of video files has a significant impact on the quality metrics of Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). In general, videos with higher resolutions tend to have lower MSE values, higher PSNR values, and SSIM values closer to 1. This suggests that higher-resolution videos better preserve image quality. The results showed that the resolution of video files has a significant impact on the quality metrics of MSE, PSNR, and SSIM. However, there is a trade-off between resolution and embedding time. While higher-resolution videos generally tend to preserve image quality better, they require slightly longer embedding times, as shown in Figure 5 (d). This trade-off between embedding time and image quality preservation is evident in the data. For example, the videos "Sunset.mp4" and "Nature.mp4" with higher resolutions have relatively higher embedding times compared to lower-resolution videos.



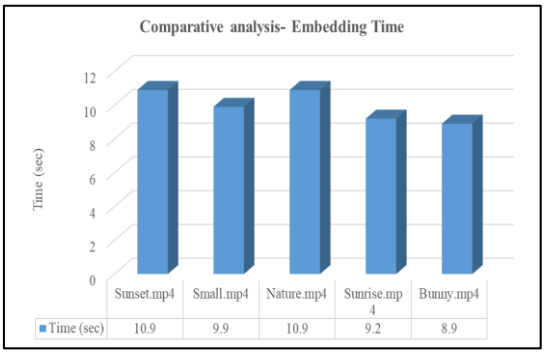
(a)



(b)



(c)



(d)

Figure 5. Comparison of Video Quality Metrics for Different Videos (a) MSE, (b) PSNR (c) and Execution Time (d)

5.3. Comparison of Payload

It is important to note that there is a trade-off between relative payload and image quality metrics. As the embedding rate increases, the quality of the image decreases. However, the number of data bits that can be concealed in each pixel of the image increases with the embedding rate. To evaluate the effectiveness of the proposed method, the system conducted a comprehensive analysis by employing payloads of varying sizes: 216 bits, 608 bits, 1816 bits, 4880 bits, and 9480 bits.

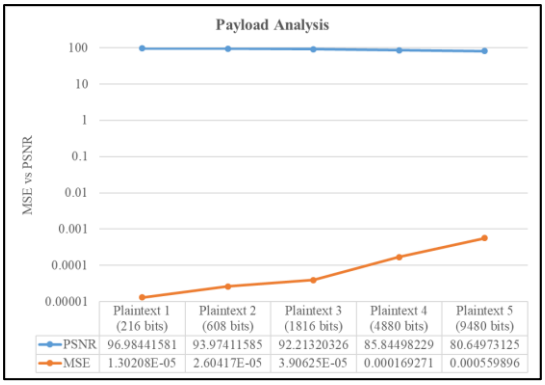


Figure 6. Comparison of Payload Capacity for Proposed Method

As shown in figure 6, the PSNR value decreases as the payload size increases. This is because the larger the payload size, the more distortion is introduced into the cover medium. The MSE value also increases as the payload size increases. This is because the larger the payload size, the more noticeable the distortion is. The results of the experiments showed that the hidden message had no significant impact on the quality of the video, even when the message was large. In fact, the number of frames used to embed data can affect the embedding process. Using more frames means that there is more data that can be hidden in the video. This can be beneficial if the system needs to hide a large amount of data. However, it can also lead to increased distortion, as the algorithm will need to modify more pixels in order to hide the data.

6. Conclusion

This paper presents a method for data hiding in a video by utilizing the Least Significant Bit (LSB) method and improving it by utilizing the Knight Tour algorithm for concealing the data inside the video file and using the AES encryption method for encrypting the secret message. The system's imperceptibility was evaluated using the PSNR, MSE, and SSIM metrics. The results showed that the PSNR and SSIM values increased with the number of frames, but decreased with the payload size. However, the embedding and extraction processes of the proposed method take longer than the simple LSB and AES-LSB methods. This is because the proposed method uses AES encryption and the knight's tour algorithm, prior to hiding the data, in order to improve security. In conclusion, the experimental results confirmed the system's effectiveness in terms of imperceptibility and embedding capability. Further enhancements can focus on reducing execution time and improving steganalysis resistance.

References

[1] Prof. Dr. P. R. Deshmukh and Bhagyashri Rahangdale, "Data Hiding Using Video Steganography," International Journal of Engineering Research & Technology (IJERT) IJERT ISSN: 2278-0181 Vol. 3 Issue 4, April 2014.

[2] A. Kumar Singh, Juhi Singh, Dr. Harsh Vikram Singh, "Steganography in Images Using LSB Technique," International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 5 Issue 1 January 2015.

[3] M. Ramalingam, "Stego Machine- Video Steganography using Modified LSB Algorithm," February 26, 2011, Journal article Open Access.

[4] Ahmad M. Odat1 and Mohammed A. Otair, "Image Steganography using Modified Least Significant Bit," Indian Journal of Science and Technology, Vol 9(39), DOI: 10.17485/ijst/2016/v9i39/86878, October 2016.

[5] A. Singh and H. Singh, ''An improved LSB based image steganography technique for RGB images,'' In 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1-4. IEEE, 2015.

[6] V. Ramesh, S. Kulkarni, and M. Ingle, "Android Application Development for Secure Data Transmission using Steganography," Transactions on Networks and Communications Vol.3, No.3, 2015, p.39.

[7] Gupta, Hemant, and Setu Chaturvedi, "Video steganography through LSB based hybrid approach," International Journal of Computer Science and Network Security (IJCSNS) 14, no. 3 (2014): 99-106.

[8] Shafna P K, "An Improved Video Steganography Technique Using Pixel Value Differencing and AES Cryptography," Computer science and engineering, KMEA College of engineering, India 2(10): October, 2015.

Investigation the Effect of Injury Severity Factors Based On Mutual Information Approach

Mya Thin Khaing, Yawai Tint
University of Computer Studies (Magway), Myanmar
myathinkhing@ucsmgy.edu.mm, yawaitint@ucsmgy.edu.mm

Abstract

Road traffic injuries are the main cause of death for children and young adults aged 5 to 29 years. An estimated 1.3 million people die each year as a result of road traffic crashes. Most nations lose 3% of their gross domestic product to road traffic crashes. Understanding the factors that influence injury severity is crucial for developing targeted interventions and prevention strategies. So, this system is intended to explore the correspondence of factors on accident data analysis based on Mutual Information (MI) approach. The selected group of factors is used to predict the injury severity. The single-vehicle crash data is obtained from the Highway Accident Information Management System, Thailand (HAIMST). In this paper, mutual information (Information Theory) approach is used to generate for maximum relevant factors and comparing the different group of factors based on MI approach that causes injury severity level such as no injury, minor, major and fatal.

Keywords: Mutual Information, Crash Severity, Factor Analysis.

1. Introduction

Injuries, both accidental and intentional, represent a significant global public health concern, exacting a substantial toll on individuals and societies. Their impact is far-reaching, effecting not only the health and well-being of those injured but also straining healthcare systems and imposing a substantial economic burden. Consequently, understanding the determinants of injury severity is of paramount importance in the realm of injury research and prevention.

Injury severity is a complex and multifaceted outcome that results from the interplay of various factors. These factors can encompass a wide range of variables, including demographic characteristics, the nature of the injury, and contextual circumstances. However, comprehensively unraveling the relationships between these factors and injury severity presents a formidable challenge.

To address this challenge, this system employs a novel and data-driven approach by harnessing the power of mutual information. Mutual information is a statistical measure rooted in information theory, which provides a rigorous framework for quantifying the strength of association between variables. So, this system is used for the investigation about the effect of factors in accidents database. Data on traffic accidents is gathered by Thailand's DOH (Department of Highway). In this paper, mutual information (based on conditional entropies) is utilized to precisely estimate the impact of accident outcome variables like injury type and severity. According to the mutual information (Information Theory), this system uses the maximum relevant factors to design for injury severity prediction.

2. Related Works

T. Boonyoo, T. Champahom, and V. Ratanavaraha [1] used structural equation modeling (SEM) to analyze the elements that determine the severity of rear-end crashes in 2022. Structural Equation Modelling (SEM) results showed that the driver component had the highest impact on accident severity, followed by the road and environmental factors. After gathering pertinent information, this study highlights stakeholder groups' crucial roles in road design, maintenance, and driver instruction. T. Champahom, S. Jomnonkwao, and V. Chatpattananan explored measures to reduce rear-end collisions and deadly rear-end crashes in 2019 [2]. Rear-end incidents have been determined to result in the greatest number of

fatalities among crash types on Thai roadways. The goal of this system was to identify strategies for reducing rear-end collisions, particularly deadly ones. The complex link between the variables in the large amount of data was analyzed using the Classification and Regression Tree (CART).

Based on the previous above researches, the proposed system aims to investigate the effect of factors on crash severity. This paper demonstrated that in the domain of accidentology by selecting the most relevant and informative factors that explain injury severity in a large dataset.

3. Proposed System

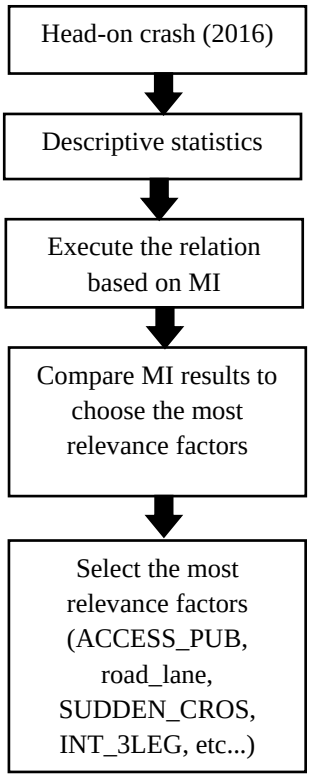


Figure 1. Process Diagram of the Proposed System

Process diagram of the proposed system is shown in Figure 1. In this system, the following steps are included as follows:

- Step 1: Extract the injury severity data (Head-On Crash)
- Step 2: Extracted data are used for data preprocessing (missing value, outlier and normalization)
- Step 3: Extract the most significant factors based on the correlation results
- Step 4: Select the maximum relevance factors from dataset by using Mutual Information (Information Theory).

4. Methodologies of the System

In this system, descriptive statistic will be conducted for dataset, significant factors will be revealed by using correlation. Most relevant factors are extracted based on mutual Information approach which is discussed in the following section.

4.1. Information Theory

Information theory is a branch of mathematics and computer science that deals with the quantification and transmission of information. It has found applications in various fields, including communication systems, cryptography, data compression, and statistics. At the core of information theory lie the concept of entropy and its various derivatives, one of which is mutual information.

4.2. Mutual Information (MI)

Relationships are described in terms of uncertainty via mutual information. The degree to which knowledge of one quantity reduces uncertainty about the other is measured by the mutual information (MI) between the two quantities. The advantage of mutual information is that it can detect any kind of relationship (non-linear relationships), while correlation only detects linear relationships. Mutual information is shown in Figure 2.

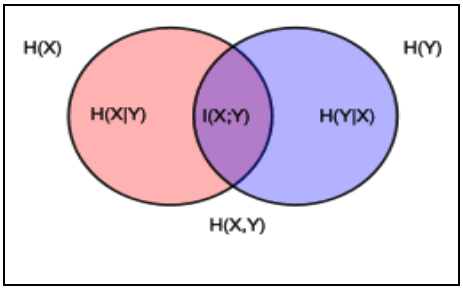


Figure 2. Mutual Information

Contrary to correlation, which can only identify linear relationships, mutual information

can identify any type of relationship (non-linear interactions).

$$I(X, Y) = H(X) + H(Y) – H(X,Y) \quad (1)$$

X is the independent variables which can be explained to injury severity outcomes and Y is the dependent variables. The independent variable(X) is the variable the experimenter manipulates or changes, and is assumed to have a direct effect on the dependent variable. The dependent variable(Y) is the variable being tested and measured in an experiment, and is 'dependent' on the independent variable.

5. Road Traffic Injuries Data

This system's data collection process comprised acquiring both original data and DOH (Department of Highway) data regarding head-on crashes. For this paper, data for analysis of highways in Thailand was drawn from 2016. In original dataset, there are 150 variables that are used to compute the significant factor based on correlation. Considering the following chart of traffic report, the traffic death rates are slightly increased in 2016. For sample data,Hid(877019),kilometer(257700),MAIN(0),PARAL(0),CONSTRUC(0),road_lane(4),Road_direction(1),NO_MEDI(0),FLUSH(0),RAISED(1), DEPRESS(0),BARRIER(0),CON_PAVE(0),STRAIGHT(1),SLOPE(0),INTERSECTION(0),INT_4LEG(0),INT_3LEG(),INT_ROUND(0),INT_RAMP(0),MEDI_OPEN(1),ACCESS_PUB(0),ACCESS_GOVE(0),etc. Trend in reported road traffic deaths in Thailand is shown in Figure 3.

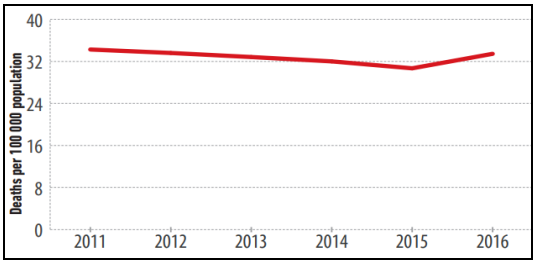


Figure 3. Trend in Reported Road Traffic Deaths in Thailand

Road traffic data is shown in Table 1. Data is divided into five factors. They are Driver factors (EXCESS_SPEED,SUDDEN_CROS,IRLEGAL_OVERTAKE, NON_SIGNAL, OVER_LOAD, DRUNK, FALL_ASLEEP), Road factors (MAIN,PARAL, CONSTRUC, road_lane, road_direction, NO_MEDI, FLUSH, RAISED, DEPRESS, BARRIER, CON_PAVE, STRAIGHT, SLOPE, INTERSECTION, INT_4LEG, INT_3LEG, INT_ROUND, INT_RAMP, MEDI_OPEN, ACCESS_PUB, ACCESS_GOVE, CROSS_WALK, BRIDGE, TUNNEL, RAIL_CROSS, INTER_UTURN), Environmental factors (WET_SURF, BROKE_PAVE, BAD_WEATHER, DAY, NIGHT_LIGHT, DARKNESS, MUNICIPAN, BANGKOK, URBAN), Rear-end factors (PREDESTRAIN,REAR_END,SIDESWIPE, SINGLE_C, HEAD_ON, DATETIME) and Crash size severity factors (No_INJURY,FATAL_CRASH,SEVERE_CRASH, MINOR_INJ_CRASH).

Table 1. Road Traffic Data

Independent Variables	**Dependent Variables**
Hid, kilometer, MAIN, PARAL, CONSTRUC, road_lane, road_direction, NO_MEDI, FLUSH, RAISED, DEPRESS,BARRIER, CON_PAVE, STRAIGHT, SLOPE, INTERSECTION, INT_4LEG, INT_3LEG, INT_ROUND, INT_RAMP, MEDI_OPEN, ACCESS_PUB, ACCESS_GOVE, CROSS_WALK, BRIDGE, TUNNEL, RAIL_CROSS, INTER_UTURN, WET_SURF, BROKE_PAVE, BAD_WEATHER, DAY, NIGHT_LIGHT, DARKNESS, EXCESS_SPEED, SUDDEN_CROS, IRLEGAL_OVERTAKE, NON_SIGNAL, OVER_LOAD, DRUNK, FALL_ASLEEP, PREDESTRAIN, REAR_END, SIDESWIPE, SIGNAL_C, HEAD_ON,,datetime, time,	No_INJURY, FATAL_CRASH, SEVERE_CRASH, MINOR_INJ_CRASH

MUNICIPAN, BANGKOK, URBAN	

Data description about independent and dependent variables are shown in Table 2 and 3.

Table 2. Data Description about Independent Variables

Independent Variables	Description	Value	
		1	0
Hid			
kilometer	Mater		
MAIN	Crash on an inner traffic lane	Yes	No
PARAL	Crash on an outer traffic lane	Yes	No
CONSTRUC	The road is under maintenance/ work zone		
road_lane	Number of Land	Count Variable	
road_direction	Direction of road		
NO_MEDI	Road was not divided by median island	Yes	No
. . .			
URBAN	Crash occurred on an urban area	Yes	No

Table 3. Data Description about Dependent Variables

Independent Variables	Description
No_INJURY	No Injury
FATAL_CRASH	At least one person (driver or passenger) killed (within 30 days) by injuries sustained in the crash
SEVERE_CRASH	At least one person injured and admitted to hospital but no fatalities
MINOR_INJ_CRASH	At least by one person requiring medical care but no fatalities or injuries requiring hospitalization

6. Implementation of the System

This system is proposed as the data analysis system to investigate the effect of injury severity factors based on mutual information approach. This system is implemented by using "Python" programming language.

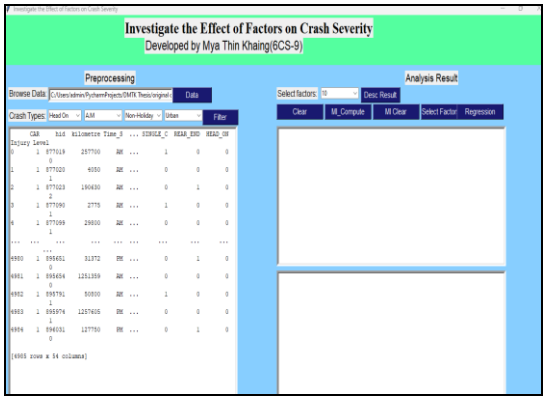


Figure 4. Road Traffic Data Extraction Process

Road traffic data extraction process is shown in Figure 4 and mutual information production process is shown in Figure 5. Mutual information results are shown in Figure 6.

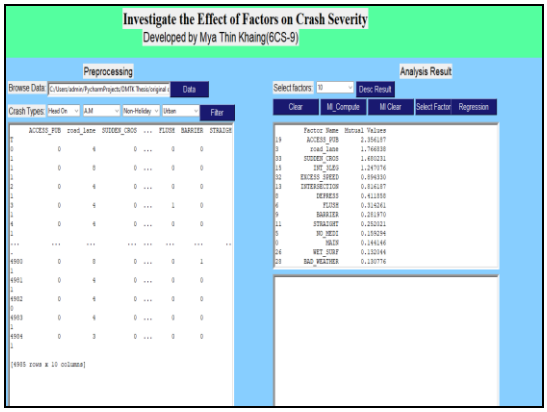


Figure 5. Mutual Information Production Process

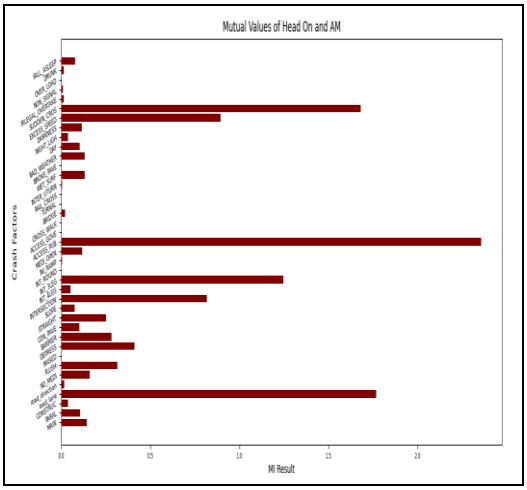


Figure 6. Mutual Information Result

7. Results and Discussion

In this section, this system describes the data description result and mutual information results with discussion.

7.1. Data Description Result

The primary features of a dataset can be summed up and described using descriptive statistics. Examples include measures of central tendency, which reveal information about the typical value in the dataset, such as mean, median, and mode. Data Descriptive Result shows in the following chart for the dataset in this system. Data description result is shown in Figure 7.

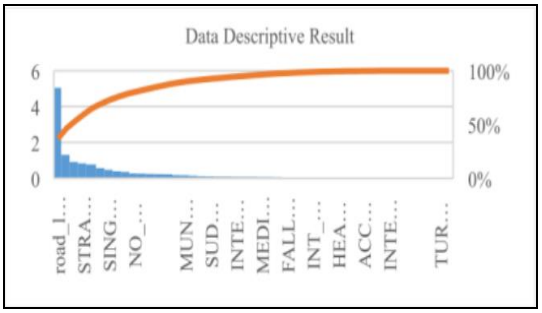


Figure 7. Data Description Result

After the descriptive statistics analysis, the system showed in the 2 tailed significance value which in this case is <0.05 (means that correlation is highly significant). The 150 variables are used to find the significant relation based on correlation results. After finding this process, 52 significant factors are extracted for analyzing injury severity outcomes based on MI approach.

7.2. Mutual Information Results

During the process of proposed system, investigating the effect of factors in injury severity is considered by different types of analysis. Based on the group of Crash types (Head-On, Rear End, Sideswipe, Pedestrian, Single-C), Time (AM, PM), Holiday and Non Holiday, Places(Urban, Municipal) comparison results, the most relevant factors are selected for injury severity prediction.

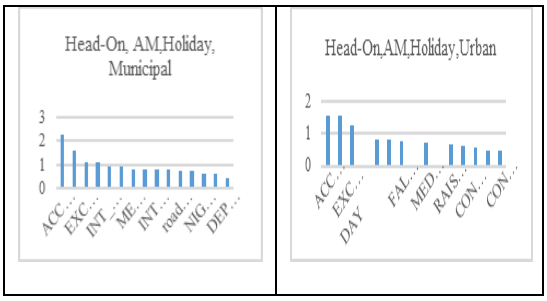


Figure 8. Comparison Municipal and Urban (am, holiday)

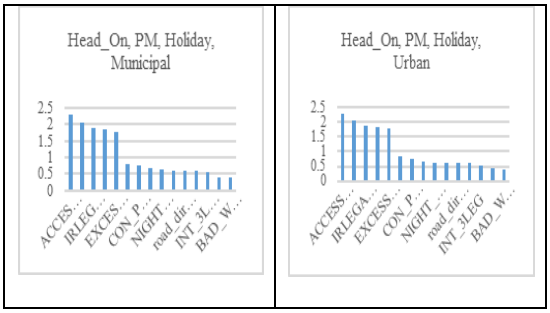


Figure 9. Comparison Municipal and Urban (pm, holiday)

Based on this results, public transport, road lane and speed factors are the highest mutual relation with injury severity outcomes.

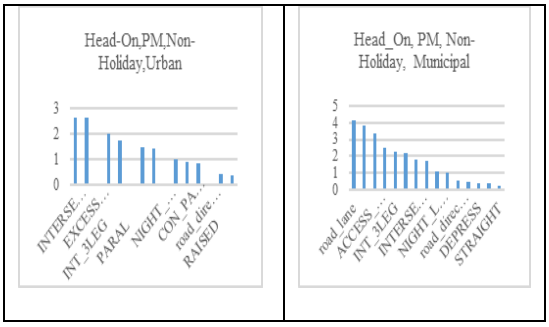


Figure 10. Comparison Municipal and Urban (pm, non-holiday)

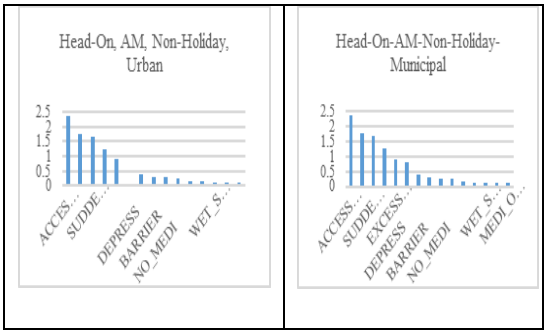


Figure 11. Comparison Municipal and Urban (am, non-holiday)

According to the results, this system described that “intersection, road barrier and paral” are the most relevance factors in night time crash at non holiday in Thailand.

In this analysis, driver behavior factors (fall asleep, excess speed) are emerged in the relevant factors based on MI approach. Moreover, road condition factors (barriers, paral and road lane) are the highest common factor in different types of based group analysis. Therefore, road condition and driver behavior factors are the most relevant factors for interpreting the injury severity outcomes in Head on Crash of Thailand.

8. Conclusion

The factors that can effect the seriousness of injuries in automobile accidents are presented in this paper. The methodology is offered for the use of an effective framework that permits the research of potential influencing factors without restriction on the type of variable. The ideal method for explaining the outcome descriptor is to choose factors using MI that produce groupings of factors with a minimal size and no repetition. One significant benefit of this strategy is its inherent ability to handle collinearity and nonlinearity. This attribute is especially helpful when dealing with accidents because traffic data frequently demonstrates a strong association between many factors (such as crash type, unstable time, and geography).

References

[1] T. Boonyoo, T. Champahom and V. Ratanavaraha, "Analysis of Factors Effecting Rear-End Crash Severity using Structural Equation Modelling", *Suranaree Journal of Science & Technology*, vol. 29, no. 1, pp. 1-12, 2022.

[2] T. Champahom, S. Jomnonkwao and V. Chatpattananan, "Analysis of Rear-End Crash on Thai Highway: Decision Tree Approach", *Journal of Advanced Transportation*, 2019.

[3] T. L. Deepika Roy, P. Deepa and K. Madhavilatha, "Road Accident Analysis using Linear Regression", *International Journal of Creative Research Thoughts (IJCRT)*, vol. 9, no. 6, 2021.

[4] K. Kameshwaran and K. Malarvizhi, "Survey on Clustering Techniques in Data Mining", *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2272-2276, 2014.

[5] G. Gandhi and R. Srivastava, “Review Paper: A Comparative Study on Partitioning Techniques of Clustering Algorithms”, *International Journal of Computer Applications*, vol. 87, no. 9, 2014.

[6] Bing Liu, *Web Data Mining*, Department of Computer Science, University of Illinois, USA, 2007.

[7] Dr. Jon Starkweather and Dr. Amanda Kay Moske: *Multinomial Logistic Regression*, 2017.

[8] Cheng Hua, Dr. Youn-Jeng Choi, Qingzhou Shi, *Companion to BER 642: Advanced Regression Method*, 2021.

[9] Adam Hayes, *Descriptive Statistics: Definition, Overview*, March 21, 2023.

[10] Mathilde Mougeot, Robert Azencott, *Injury Severity Analysis based on mutual information for in depth investigation of accident database*, July 9, 2010

[11] A. Mukherjee and P. Mitra, "Prediction Cyclone Landfall using Mutual Information and Dilated Recurrent Neural Network", Cambridge University Press, *Environmental Data Science*, 2022.

Detection and Classification of Paddy Leaf Disease based on the Combination of OTSU's Method and Support Vector Machine (SVM)

May Phyu Phyu Aung, Phyo Phyo Wai
University of Computer Studies (Magway), Myanmar
mayphyuphyuaung@ucsmgy.edu.mm, phyophyowai@ucsmgy.edu.mm

Abstract

Rice is the most consumed staple food for more than half of world's population. The least expensive way to keep up with population growth is to increase the rice output. One of the major problems in rice production is the paddy leaf diseases. So, an automated system is proposed the current trends and challenges for detection and classification of paddy leaf diseases. For the image acquisition, this system is used a dataset that includes four kinds of Rice leaf diseases (Bacterial blight, Blast, Brown Spot and Tungro) from Mendeley Data. This system consists of three sub-processes that are pre-processing, feature extraction and classification. In pre-processing step, the RGB images are converted into gray scale images. Then, median filtering method is used to remove noise from the gray scale image. For the segmentation of diseased portion and non-diseased portion, OTSU thresholding method is used. In the feature extraction, gray-level co-occurrence matrix (GLCM) method is used to extract features from the segmented image. Using the extracted features, this system classifies the paddy leaf disease according to the support vector machine classifier.

Keywords: Classification, Paddy Leaf Disease, OTSU, GLCM, SVM.

1. Introduction

In Myanmar, one of the important sources of earning for human beings is agriculture. In the rainy rice planting season of 2022 Myanmar, when the rainy rice was planted for a month and a half, there was almost a month and a half of no rain, which has not been experienced in 60 years, and a water shortage was faced. Due to such a drought, the soil crust of the farmland is tight and the growth of plants, infection, the decrease in the growth rate of plants, due to the situation of buying and using high prices of fertilizers, they are not able to add enough fertilizer and use it. Disease is one of the factors that reduce rice productivity. The majority of problems are mitigated by providing certain technological resources. So, paddy leaf disease identification is one of the most important research topics in the field of agriculture. Recognizing the paddy leaf disease is essential to avoiding reductions in the amount and productivity of agricultural output. So, paddy leaf disease detection and health monitoring are very harmful to sustainable agriculture.

So, this system is proposed as the paddy leaf disease detection and classification system by using support vector machine (SVM) method. This system involves pre-processing, detection and classification the paddy leaf disease. All paddy leaf samples will be passing through the RGB calculation in which input images firstly convert gray image for reducing pixels value and filtering for removing noise. Then, the image segmentation is performed to make the image analysis easy and meaningful. For segmentation, this system uses the Otsu method to separate diseased regions from healthy regions based on intensity differences. Then, this system extracts the meaningful and informative features from images to facilitate subsequent analysis or tasks. GLCM (Gray-Level Co-occurrence Matrix) method is used for feature extraction. Using the extracted features, this system classifies the paddy leaf diseases. These diseases are "Bacterial blight", "blast", "brown spot" and "Tungro" paddy leaf disease.

The proposed system helps the farmers to classify the diseased paddy leaves and to protect the crops from further damage. This system enhances their ability to respond to changing conditions and challenges in the agricultural environment.

2. Related Works

In 2017, content based paddy leaf disease recognition and remedy prediction using support vector machine was developed by F. T. Pinki, N. Khatun and S.M. M. Islam. They used a dataset from Bangladeshi rice-producing areas. The 900 x 750 pixels image resizing is required during pre-processing. Median filtering and contrast enhancement are used to reduce noise, smooth the image, and improve the image so that distinct parts can be understood more clearly. To segment the affected area, the k-means clustering technique is employed. SVM is employed for classification [1].

In 2020, recognition and classification of paddy leaf diseases is presented by S.Ramesh, D.Vydeki. They used a dataset from the agricultural fields. The 300 x 450 pixels photos must be resized during pre-processing. RGB photos are transformed to HSV images for background removal, and binary images are retrieved to separate the sick and non-diseased portions based on hue and saturation. To segment the affected area, the k-means clustering is employed. For classification, the optimized deep neural network with Jaya optimization algorithm (DNN_JOA) is employed [2].

In 2021, Image processing and Pattern Recognition based Plant Leaf diseases Identification and Classification was developed by K. Gayatri et al. This system used contrast enhancement for pre-processing, k-means clustering for the segmentation of the diseased portion, various techniques for feature extraction, and support vector machine (SVM) for classification. The experimental results center on the identification of the four styles of diseases.

3. Proposed System Design

This system is proposed as the detection and classification of paddy leaf diseases based on the support vector machine (SVM) classifier. These are data collection, data pre-processing, feature extraction and classification.

Firstly, this system collects various types of paddy leaf diseases (Bacterial blight, Blast, Brown Spot and Tungro). After collecting the paddy leaf data images, this system performs the pre-processing step to enhance the quality of the data. In the pre-processing step, this system converts the RGB image to the gray scale image and then this system removes noise from this gray scale image by using median filtering method. System flow diagram is shown in Figure 1.

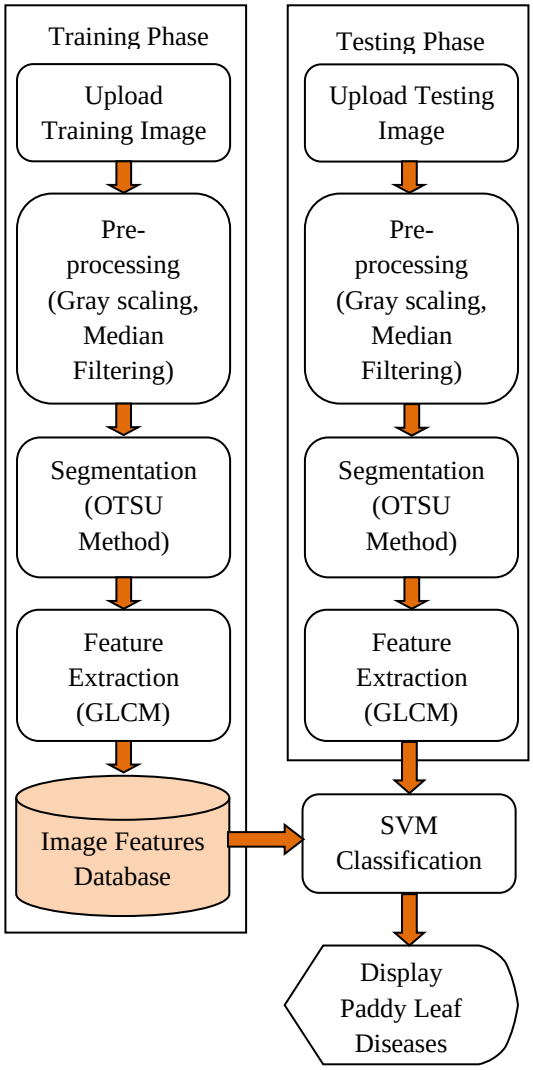


Figure 1. System Flow Diagram

To extract the paddy leaf disease region, this system performs the segmentation process according to the OTSU method. In the feature extraction step, this system extracts the relevant features from the segmented images. This system uses the gray-level co-occurrence matrix (GLCM) method to extract features.

After extracting features from the training paddy leaf images, this system classifies the testing paddy leaf image that suffers one of the paddy leaf diseases. According to the SVM classifier, this system can classify the “Bacterial Blight”, “Blast”, “Brown Spot” and “Tungro” paddy leaf disease.

4. Methodologies of the System

As a methodology, this system consits of RGB to gray scale conversion, median filtering, OTSU's image segmentation, GLCM feature extraction and SVM classifier.

4.1. Image Acquisition

This system uses the Mendeley data that includes the rice leaf disease image samples. These data are trained to detect and classify the paddy leaf disease. This data is extracted from https://data.mendeley.com/datasets/fwcj7stb8r/1. File size of Mendeley data is "171 MB". Taining paddy leaf disease data is shown in Table 1.

Table 1. Training Paddy Leaf Disease Data

Disease Types	Images	Image Type	Size
Bacterial Blight	1584	JPG	300*300
Blast	1440	JPG	300*300
Brown Spot	1600	JPG	300*300
Tungro	1308	JPG	300*300

4.1. RGB to Gray Scale Conversion

Red, Green, and Blue (RGB) are the three fundamental colors of light. In an RGB image, the color intensity for each of the three color channels is represented by each pixel value. The color intensity of each pixel in a gray scale image is represented by its single color channel. Gray scale image makes them easier to examine and process more quickly [4].

4.2. Median Filtering

To create different effects, filters are mathematical operations that are applied to each pixel or a neighborhood of pixels in a picture. For each pixel in the image (excluding border pixels), place a 3x3 window centered around the pixel. Median filtering process is shown in Figure 2.

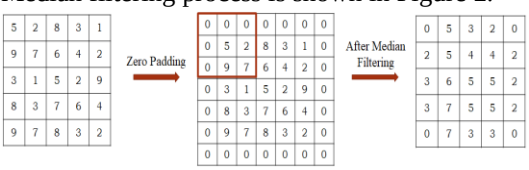


Figure 2. Median Filtering Process

The process of median filtering is as follows:

- Step 1: Collect the intensity values of all the pixels within the window.
- Step 2: Sort collected intensity values in ascending order.
- Step 3: Calculate the median value, which is the middle value in the sorted list. If the window size is odd (as in this case), the median is the value at the (n+1)/2 position, where n is the total number of intensity values.
- Step 4: Replace the original pixel value with the calculated median value.
- Step 5: Repeat steps 1-4 for all pixels in the image [5].

4.3. OTSU's Image Segmentation

Otsu method is a thresholding technique used for image segmentation. Through the process of maximizing the between-class variation of pixel intensities, it derives the ideal threshold value. By using intensity differences to distinguish between infected and healthy zones, the Otsu technique can be used to identify plant diseases. Otsu's image segmentation algorithm is as follows:

- Draw the histogram of the digital image and calculate the various intensity levels.
- Initialize ω_b,ω_f and μ_b , μ_f.
- Calculate all possible threshold values from t = 1…….n.
  - Update ω_b,ω_f and μ_b,μ_f for all possible values
  - Calculate σ_b^2 for all threshold values t.

$$\sigma_B^2(t) = (\omega_f(t).\omega_b(t)).(\mu_f(t) - \mu_b(t))^2$$

- Desired threshold relates to maximum of σ_b^2.

In Otsu's method, "ω_i" is weight, "μ" is mean and "σ_i" is standard deviation [6]. Original image, filtered image and segmented image of bacterial blight is shown in Figure 3.

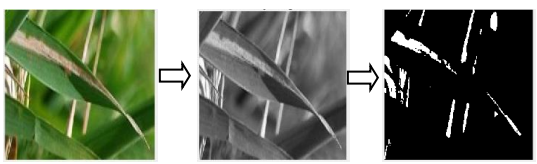


Figure 3. Original Image, Filtered Image, Segmented Image of Bacterial Blight

4.4. GLCM Feature Extraction

For efficient plant disease detection and classification, extracted features (useful elements from photographs) are fed into classifiers. Among

many features extraction method, GLCM (Gray-Level Co-occurrence Matrix) is a texture analysis technique to obtain statistical data about the spatial associations between pixels in a photo. GLCM technique computes a matrix that indicates how frequently various combinations of gray-level values occur at particular pixel offsets or displacements in a picture [7]. GLCM process is shown in Figure 4.

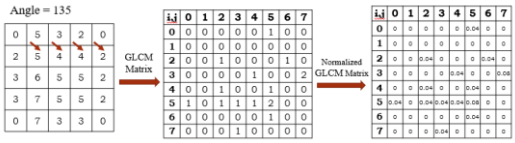


Figure 4. GLCM Feature Extraction Process

According to the GLCM feature extraction method, matrices of GLCM frequencies are a function of the angular relationship and distance between the neighboring pixels. Formally, for angles quantized to 135 intervals, the unnormalized frequencies are defined. This system extracts the 22 features. These extracted features results are shown in Table 2.

Table 2. GLCM Feature Results from the Bacterial Blight

ID	Features Name	Features Value
1	Autocorrelation	1.1937
2	Contrast	0.0206
3	Correlation	0.8415
4	Correlation1	0.8415
5	Cluster prominence	0.7106
6	Cluster shade	0.3856
7	Dissimilarity	0.0206
8	Energy	0.8537
9	Entropy	0.3321
10	Homogeneity	0.9899
11	Homogeneity1	0.9899
12	Maximum probability	0.9220
13	Sum of square	0.7316
14	Sum average	2.1358
15	Sum variance	3.5369
16	Sum entropy	0.3185
17	Difference variance	0.0266
18	Difference entropy	0.0982
19	Information measure of correlation	-0.6601
20	Information measure of correlation1	0.5293
21	Inverse difference normalized	0.9933
22	Inverse difference moment normalized	0.9959

4.5. Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised learning model that is employed for binary classification. But, they can be modified to accommodate multi-class problems. Finding the best hyperplane to divide the data points of several classes in a high-dimensional feature space is the fundamental notion behind support vector machines (SVM). Maximizing the margin between classes is the goal of SVM, which improves generalization and robustness to unknown data. Gaussian SVM involves optimizing parameters that are the regularization parameter (C), the width of the Gaussian kernel and kernel-specific parameters [8]. SVM classifier is shown in Figure 5.

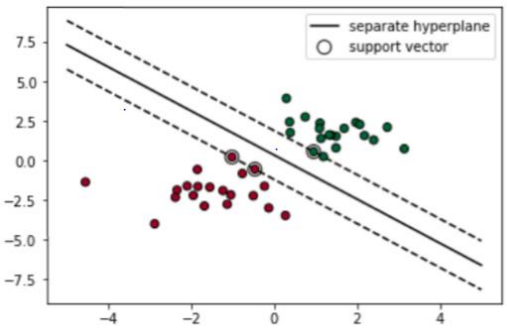


Figure 5. SVM Classifier

5. Implementation of the System

This system is proposed as the detection and classification of paddy leaf diseases based on support vector machine (SVM). This system is implemented by using MATLAB programming language. Firstly, the user must input the RGB image via “Load Image”. For this conversion, this system uses the “Grayscale”. In this system, noise can reduce from the “Median Filtering”.

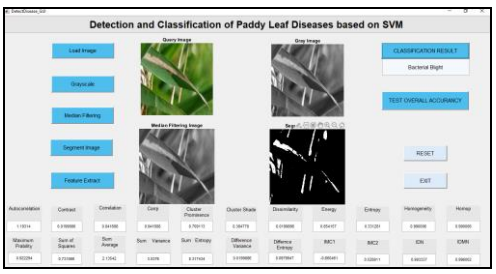


Figure 6. Implementation of Paddy Leaf Disease Detection and Classification System

Implementation of paddy leaf disease detection and classification system is shown in Figure 6. Image segmentation helps in identifying and localizing objects within an image. This system segments the filtered image by using "Segment Image". Then, this system extracts 22 features from the segmented image. To extract features, this system uses "Feature Extract" button. After extracting features, this system classifies the inputted image that suffers one of the paddy leaf diseases that are "Bacterial Blight", "Blast", "Brown Spot" and "Tungro".

6. Experimental Result of the System

To show the performance of the system, this system uses the 5-fold cross validation accuracy method. For classification, this system needs to train the paddy leaf disease data into the system. For training and testing dataset, this system uses the 800 paddy leaf disease images that include 200 "Bacterial Blight" disease image, 200 "Blast" disease image, 200 "Brown Spot" disease image and "Tungro" disease image. The file size of these data is "18.4" MB. These data are extracted from the Mendeley Data website (link: https://data.mendeley.com/datasets/fwcj7stb8r/1). This system uses nineteen features that are detail described in section 4.4 and four classes for paddy leaf disease classification. According to the Gaussian SVM, this system classifies and produces result that is one of the paddy leaf diseases.

This system calculates the true positive rate (TPR) and false negative rate (FNR) to produce the confusion matrix. Confusion matrix from Figure 9 shows the accuracy of the system. Accuracy of the system is shown in Table 3.

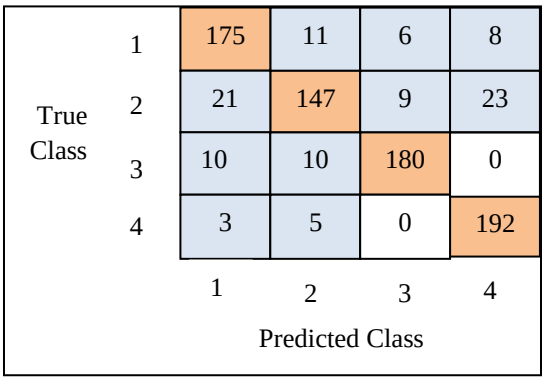


Figure 7. Confusion Matrix of the System

Table 3. Accuracy of the System

Class ID	Class Name	No of True Class	No of False Class	Accuracy (%)
1	Bacterial Blight	175	25	87.5 %
2	Blast	147	53	73.5%
3	Brown Spot	180	20	90%
4	Tungro	192	8	96%

According to the accuracy results, this system more correctly classifies the "Tungro" paddy leaf disease than other three paddy leaf diseases. Among four paddy leaf diseases, the "Blast" paddy leaf disease classification has highest error rate. The correct rate of each paddy leaf disease classification is shown in Figure 8.

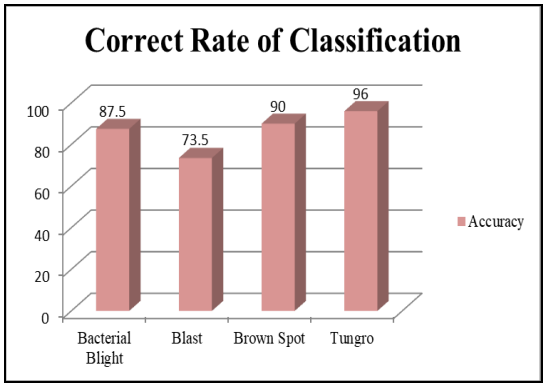


Figure 8. Correct Rate of Paddy Leaf Disease Classification

According to the paddy leaf classification results, the overall correct classification rate of the system is "86.75" and the error rate of the system is "13.25". The experimental result of the system is shown in Figure 9.

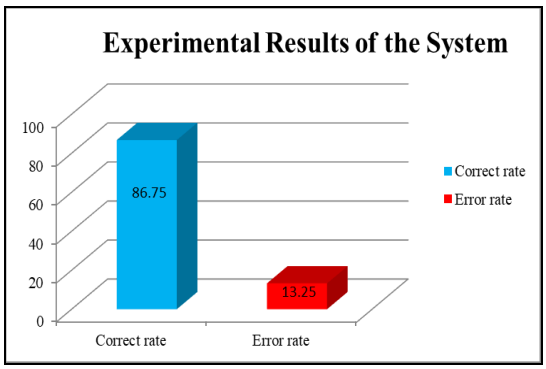


Figure 9. Experimental Results of the System

This system compares the performance of "Gaussian SVM" with other classifiers (K-nearest Neighbor (KNN) classifier and decision tree classifier). This "Gaussian SVM" classifier accuracy is the highest among other classifier. The accuracy of "Gaussian SVM" classifier is more 10% than "KNN" classifier. And, the accuracy of "Gaussian SVM" classifier is more 13.1% than "Decision Tree" classifier. Accuracy results about each classifier are shown in Table 4.

Table 4. Accuracy Results about Each Classifier

ID	Classifier Name	Accuracy (%)	
		Correct Rate	Error Rate
1	Support Vector Machine	87.6%	12.4%
2	K-Nearest Neighbor	77.6%	22.4%
3	Decision Tree	74.5%	25.5%

Performance comparison result is shown in Figure 10.

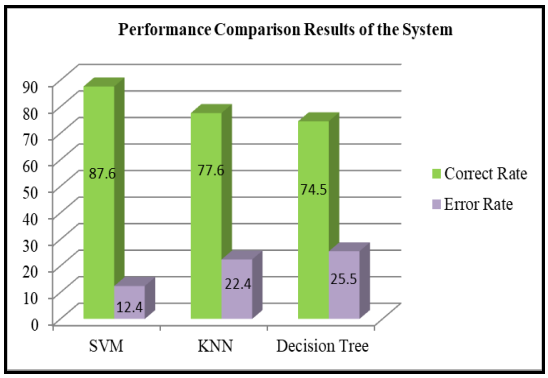


Figure 10. Performance Comparison Result

7. Result and Discussion

In this study, paddy leaf disease classification system has been developed by using OTSU's approach and Support Vector Machine (SVM). In this analysis, there are three stages for classification of the paddy leaf diseases by acquiring the "Rice Leaf Disease Image Samples" from Mendely Data. According to the analysis, GLCM features are used to detect clearly to the "bacterial blight, brown spot, and tungro" diseases. However, "blast" disease are not detected accurately based on the GLCM features. GLCM features are have to analyzed the statistical texture features in images. By using the OTSU's method to separate the foreground and background of paddy lead disease images. If the foreground and background regions have equal variances, which may not be true in the "blast" and "bacterial blast" cases, leading to poor segmentation results. By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model's generalization performance of paddy leaf classification system. Based on the result, the highest accuracy of Gaussian kernel function (86.75%) is selected for classifying the four group of paddy leaf diseases. The result revealed that "tungro" disease has the highest classification rate (96%) than other diseases.

References

[1] F. T. Pinki, N. Khatun and S.M. M. Islam, "Content based Paddy Leaf Disease Recognition and Remedy Prediction using Support Vector Machine", *20th International Conference of Computer and Information Technology (ICCIT), IEEE*, pp. 22-24, 2017.

[2] S. Ramesh and D. Vydeki, "Recognition and Classification of Paddy Leaf Diseases using Optimized Deep Neural Network with Jaya Algorithm", *Information Processing in Agriculture, Elsevier*, pp. 249-260, 2020.

[3] K. Gayatri et al, "Image Processing and Pattern Recognition based Plant Leaf Diseases Identification and Classification", *ICMAICT*, pp. 1-10, 2021.

[4] C. Saravanan, "Color Image to Grayscale Image Conversion", *Second International Conference on Computer Engineering and Applications, IEEE*, pp. 196-199, 2010.

[5] J. Micek and J. Kapitulik, "Median Filter", *Journal of Information, Control and Management System*, pp. 51-56, 2003.

[6] C. V. V. S. Srinivas, M. V. R. V. Prasad and M. Sirisha, "Remote Sensing Image Segmentation using OTSU Algorithm", *International Journal of Computer Applications*, pp. 46-50, 2019.

[7] S. A. Alazawi, N. M. Shati and A. H. Abbas, "*Texture Features Extraction based on GLCM for Face Retrieval System*", Periodicals of Engineering and Natural Sciences, ResearchGate, pp. 1459-1467, 2019.

[8] C. A. Omar, A. S. Dora and A. A. T. Jose, "Supervised Learning using Support Vector Machine Applied to Sentiment Analysis of Teacher Performance Satisfaction", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 1, pp. 516-524, 2022.

[9] sethy, prabira Kumar (2020), "Rice Leaf Disease Image Samples", Mendeley Data, V1, doi: 10.17632/fwcj7stb8r.1

Phishing URL Detection System Using Ensemble Learning Methods

Theint Theint Win, Tin Zar Thaw
University of Computer Studies, Yangon, Myanmar
theinttheintwin@ucsy.edu.mm, tinzarthaw@ucsy.edu.mm

Abstract

In the current year, the cyber attackers use many kinds of phishing attacks, such as fake websites or links, emails, and messages, to obtain private information. The phishing detection system is very important to control cybercrimes. The proposed system applies the ANOVA feature selection method based on the UCI dataset. The ANOVA selects the most relevant twenty-six features from thirty features of the dataset. The system combines three machine learning classifiers: Random Forest, Gradient Boosting, and Support Vector Machine for improved performance using two ensemble learning methods: hard voting and soft voting. And then three classifiers are supported to produce a more accurate Phishing URL Detection System. In this paper, the testing accuracy of the UCI dataset is obtained at 96.88% for soft voting and 96.65% for hard voting, displaying the performance of the proposed system. It is implemented by the Python programming language to support protecting users from phishing threats. The proposed model would be practical for helping cybersecurity specialists and the public accurately detect phishing attacks.

Keywords: Phishing attacks, Cybercrimes, ANOVA, UCI Phishing dataset, Ensemble learning, Random Forest, Gradient Boosting, Support Vector Machine, Hard voting, Soft voting, Cybersecurity

1. Introduction

Social engineering means that many criminals target users' weaknesses. Phishing is the most famous form of social engineering attack. In today's digital age, the rapid evolution of the web is influenced by the expansion of online services such as online software downloading and clicking, payment, and social websites [1]. Therefore, the ensemble learning methods are an effective approach for detecting and preventing phishing attacks. This attack tricks users to steal sensitive personal information, IDs, and account numbers.

The proposed system utilizes ensemble learning techniques to classify phishing or legitimate. The system uses three classifiers, which are Random Forest, Gradient Boosting, Support Vector Machine. By using ANOVA feature selection, the selected optimal features can lower the time expense in model training and reduce memory consumption. Effectively classifying phishing URLs is important to recognize and prevent phishing attacks. So, the proposed system achieves stronger performance and supports users to protect from against phishing attacks.

2. Related Works

A wide range of research and studies have been conducted on phishing detection.

In paper [2], the author employs UCI dataset 4898 phishing websites and 6157 legitimate websites. It combines predictions from four different machine learning algorithms and employs a novel weighted soft voting technique based on Kappa statistics. The research shows the effectiveness of dynamic weighting in enhancing phishing websites detection. The approach in [3] extracted only sixteen important features from the URL string. The study utilized Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), and Artificial Neural Network (ANN) as classifiers for malicious URL detection. The dataset collected by Phish Tank consists of 11,033 malicious and 5,004 benign. Evaluation metrics obtained the accuracy of 88%, 87%, and 88%, respectively. In paper [4], ensemble learning approach is implemented using multiple classifiers, including LR, Bagging, RF, K-NN, DT, SVM, and Adaboost for the detection of

phishing websites. The study also introduces a meta-learner and stacking model to improve detection. Ensemble-based meta-learners for stacking are more effective than traditional models for phishing detection.

3. Background Theory

The first crime of phishing attack was posed via emails to America Online (AOL) users since the early to mid-1990s [5]. The primary steps of a phishing attack are as follows:

Step 1 Contextualization:

Attackers collect customers' or users' detailed information lists.

Step 2: Impersonation

Attackers impersonate legitimate or authorized organizations to deceive victims.

Step 3: Communication

Attackers communicate with users via email link or SMS to steal sensitive data.

Step 4: Exploitation

Attackers gain the user's credential after a successful intrusion. And then they sell users' information to the dark web [6].

3.1. Dataset

The proposed system employs a phishing detection dataset sourced from the UCI Machine Learning Repository. The dataset comprises thirty features and contains over 11,000 instances, with a total of 11,055 rows. To ensure the model is well-trained and properly evaluated, the dataset is split into 80% for training and 20% for testing.

Specifically, the dataset includes 4,897 instances of phishing URLs and 6,158 instances of legitimate URLs. These labels are encoded as -1 for phishing and 1 for legitimate, facilitating binary classification tasks.

3.2. ANOVA Feature Selection

ANOVA is a popular numerical feature selection technique. The average values of distinct classes are calculated to select significant features by statistical method. This method filters irrelevant features from the dataset but not related target values. The ANOVA-test, F-test, or F-statistic is a determined statistical test for between two different classes [7]. In this study, ANOVA has chosen twenty-six features from thirty features of the UCI dataset. In the selection, the most valuable features were selected after conducting experiments. The selected twenty-six features performed better than the thirty features. The highest twenty-six values of the F-statistic are shown in figure 1. The dataset rules of the selected features are as follows [8]:

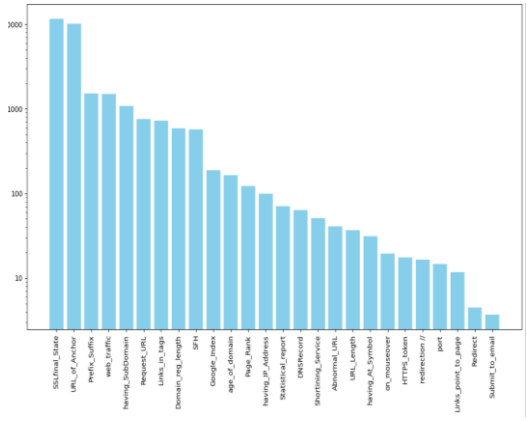


Figure 1. Twenty-six features

Having IP Address: If the IP address is contained in the URL, the feature value is -1 for "phishing" and else 1 for "legitimate."

URL Length: If the length of the URL is less than 54, the feature value is 1 for "legitimate," else if it is between greater than 54 and less than 75, the feature value is 0 for "suspicious," and else the value is -1 for "phishing."

Shortening Services: When a tiny, short URL is contained in the URL, the feature label is -1 for "phishing"; otherwise, the feature label is 1 for "legitimate".

having "@" symbol: When the "@" symbol presents in the URL is -1 for "phishing," otherwise 1 for "legitimate."

double slash redirecting: When the last occurrence of a double slash in the URL is greater than 7, the feature result is -1 for "phishing," otherwise 1 for "legitimate."

Prefix_Suffix "-": When the domain name contains the "-" symbol in the URL, the feature is -1 for "phishing," otherwise 1 for "legitimate".

having_Sub_Domain: If the dot in the domain is equal to 1, the feature value is 1 for "legitimate," else if the dot in the domain is equal to 2, the feature value is 0 for "suspicious," otherwise the feature value is -1 for "phishing."

SSLfinal_State: If the HTTPS and issuer are trusted and the certificate age is greater than or equal one years, the feature value is 1 for "legitimate," else if the HTTPS and issuer are not

trusted, the feature value is 0 for "suspicious," otherwise -1 for "phishing."

Domain_registeration_length: When the domain expires in less than or equal 1 year, the feature label is -1 for "phishing," otherwise 1 for "legitimate."

Non-Standard Port: By the rule, HTTPS and HTTP standard ports, that is, 443 and 80 ports, should be opened. When the non-standard port opens -1 for "phishing" otherwise 1 for "legitimate".

HTTPS_token: When the HTTP token is in the domain part of the URL, the feature label is -1 for "phishing," otherwise 1 for "legitimate."

Request URL: If the percentage of the request is less than 22, the feature value is 1 for "legitimate," else if the percentage of the request is between greater than or equal to 22 and less than or equal to 61, the result is 0 for "suspicious," otherwise -1 for "phishing."

URL of Anchor: If the percentage of the anchor URL is less than 31, the feature value is 1 for "legitimate", else if the percentage of the anchor URL is between greater than or equal to 31 and less than or equal to 67, the feature value is 0 for "suspicious," otherwise -1 for "phishing."

Link in <meta>, <script>, <link> tags: If the percentage of the <link>, <meta>, and <script> tags is less than 17, the feature is 1 for "legitimate", else if the percentage of <link>, <meta>, and <script> tags is between greater than or equal to 17 and less than or equal to 81, the feature is 0 for "suspicious," otherwise -1 for "phishing."

Server Form Handler: If the SFH is "about: blank" or empty, the feature value is -1 for phishing; else, if the SFH is a different domain name, the feature value is 0 for "suspicious," otherwise 1 for "legitimate."

Submitting to email: When the mail or mail to () function uses the submit form, the feature value is -1 for "phishing," otherwise 1 for "legitimate."

Abnormal URL: When the host name is not included in the URL, the feature value is -1 for "phishing," otherwise 1 for "legitimate."

Redirect: If the redirect page is less than or equal to 1, the feature label is 1 for "legitimate," else if the redirect page is between greater than or equal to 2 and less than 4, the feature label is 0 for "suspicious," otherwise -1 for "phishing."

on_mouse_over: When "onMouseOver" changes the status bar, the feature value is -1 for "phishing," otherwise, it does not change the status bar; the feature value is 1 for "legitimate."

Age of domain: When the domain age is greater than or equal to 6 months, the feature label is 1 for "legitimate," otherwise -1 for "phishing."

DNS Record: When the domain name is not recorded, the feature value is -1 for "phishing," otherwise, 1 for "legitimate."

Web traffic: If the website rank is less than 100,000, the feature value is 1 for "legitimate," else if the website rank is greater than 100,000, the feature value is 0 for "suspicious," otherwise -1 for "phishing."

Page rank: When the page rank is less than 0.2, the feature label is -1 for "phishing," otherwise, 1 for "legitimate."

Google Index: When the webpages are indexed in Google, the feature value is 1 for "legitimate," otherwise -1 for "phishing."

Links Pointing to Page: If the links pointing to the page are equal to 0, the feature value is -1 for "phishing," else if links pointing to the page are between greater than 0 and less than or equal to 2, the feature value is 0 for "suspicious", otherwise, 1 for "legitimate."

Statistical report: If the top phishing IP or domain is included in the report, the feature result is -1 for "phishing," otherwise, 1 for "legitimate."

3.3. Machine Learning Classifiers

Machine learning combines computational analysis of algorithms and statistical techniques without being specifically programmed. The benefit of the machine learning algorithm is that once it learns data, it works automatically. The distinct algorithms solve the issues based on machine learning [9].

The proposed system uses effective three classifier models: Support Vector Machine, Gradient Boosting, and Random Forest that are detailed explained as follows.

3.3.1. Support Vector Machine Classifier

A machine learning algorithm, Support Vector Machine, is also employed for both classification and regression problems. Support Vector Machine works well in high-dimensional data. Its purpose is to determine the distance between the closest points and the hyperplane. In a feature space, the hyperplane or decision

boundary is utilized for dividing the data points of distinct classes [11]. So, this algorithm finds the optimal hyperplane.

3.3.2. Gradient Boosting Classifier

A machine learning algorithm, Gradient Boosting, is also employed for both classification and regression problems, and it is a boosting ensemble learning algorithm. Gradient Boosting makes many types of weaker learners into strong learners. It includes a sequential decision tree to build a training model, and a new model corrects errors from previous models. The decision tree and linear models are employed widely as based learners in Gradient Boosting [10].

3.3.3. Random Forest Classifier

A machine learning algorithm, Random Forest, is employed for both regression and classification problems. It is a bagging ensemble learning algorithm. The bagging method is the more accurate performance of machine learning. Random Forest contains many numbers of decision trees [11]. The final prediction is obtained by a majority vote or average values from each decision tree's results. For classification, it determines the majority value. For regression, it determines the average value from each tree.

3.3.4. Two Types of Ensemble learning Techniques

Many types of based models are combined into an ensemble learning method to perfect models and accurate accuracy [11]. There are two voting classifiers in ensemble learning: soft voting and hard voting. The proposed system employs two voting classifiers based on three classifier models.

Soft Voting: In soft voting classifier, the final results are determined by the average probabilities from different classifiers. This technique sums the average probabilities from each class and then selects the highest sum values for final predictions [11].

Hard Voting: In hard voting classifier, final results are determined by combining different model classifiers through the highest number of votes. The maximum number of voting results from each classifier is chosen for the final prediction.

4. Design of the Proposed System

In this section, the Phishing URL Detection system employs UCI dataset total numbers of 11055 rows and twenty-six features. There is phishing (4897) rows or legitimate (6158) rows. The result labels are -1 or 1. -1 represents a phishing label, and 1 represents a legitimate label. This dataset divides into a training dataset for 80% and a testing dataset for 20%. The ANOVA feature selection method selected more relevant features for better system performance. The system predicts the UCI dataset for training and testing performance. The proposed system applies ensemble learning techniques. Ensemble learning methods are associated with multiple machine learning models. Two ensemble learning approaches: hard voting and soft voting are employed in the system. Hard voting predicts majority voting, and soft voting predicts average probability results from multiple classifiers. These two methods are based on three machine learning classification models: Random Forest, Gradient Boosting, and Support Vector Machine. This system implements the Python programming language, which has appropriate library functions. The proposed system purposes to enhance accuracy and support protecting against phishing threats.

4.1. Overview of the Proposed System

In this part, there are three flow diagrams. In figure 2, there are many steps in the system flow diagram for training.

Step 1: The system starts for the initial stage.

Step2: The next step inputs the UCI phishing detection dataset.

Step 3: After step 2, the dataset is selected with relevant twenty-six features by using the ANOVA feature selection method.

Step 4: The system divides into two parts of the dataset. 80% for training and 20% for testing dataset.

Step 5: After splitting the dataset, the system builds three classifier models: Random Forest (RF), Gradient Boosting (GB), and Support Vector Machine (SVM).

Step 6: And then, appear three classification models based on the training dataset.
Step 7: At last, the system continues to use 20% of the testing dataset for evaluation.

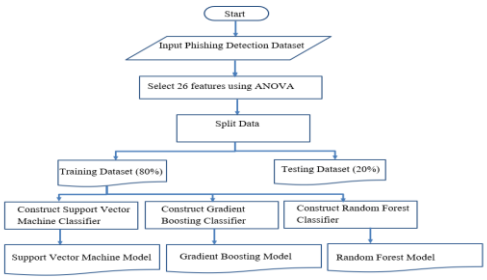


Figure 2. System Flow Diagram for Training

In figure 3, there are many steps in the system flow diagram for soft voting testing.
Step 1: The system starts for the initial stage.
Step 2: The next step inputs the UCI phishing detection testing dataset.
Step 3: After step 2, the system predicts with three classifier models: Random Forest (RF), Gradient Boosting (GB), and Support Vector Machine (SVM).
Step 4: The probabilities results achieved from three classifier models.
Step 5: After that, these are calculated average probabilities from three models.
Step 6: And then, the system chooses the maximum probability result.
Step 7: Finally, the system displays the performance result and end stage.

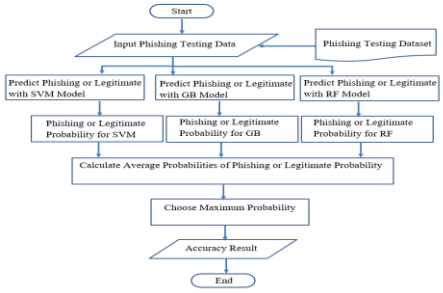


Figure 3. System Flow Diagram for Testing (Soft Voting)

In figure 4, there are many steps in the system flow diagram for hard voting testing.
Step 1: The system starts for the initial stage.
Step 2: The next step inputs the UCI phishing detection testing dataset.
Step 3: After step 2, the system predicts with three classifier models: Random Forest (RF), Gradient Boosting (GB), and Support Vector Machine (SVM).
Step 4: The voting results achieved from three classifier models.
Step 5: After that, these are counted as maximum votes from three models.
Step 6: And then, the system chooses the maximum voting result.
Step 7: Lastly, the system displays the performance result and end stage.

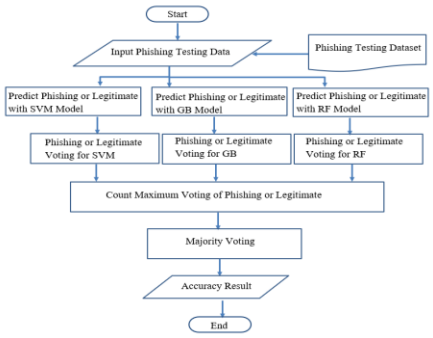


Figure 4. System Flow Diagram for Testing (Hard Voting)

5. Experimental Results

The proposed system applies five models of machine learning classifiers to achieve better accuracy. The accuracy percentage of the model is calculated as follows:

The soft voting method has the highest accuracy of 96.88% among the other methods. The hard-voting method has an accuracy of 96.65%, Random Forest has an accuracy of 96.61%, Gradient Boosting has an accuracy of 96.56%, and Support Vector Machine has an accuracy of 94.97%, respectively. Figure 5 shows the experiment results.

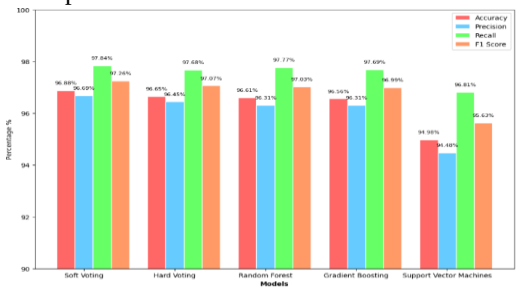


Figure 5. Experiment Results

All features are tested by using ANOVA based on ensemble learning models to enhance accuracy. After experimentation, the best features are twenty-six for five models. So, the best 26 features are displayed as a red circle in Figure 6.

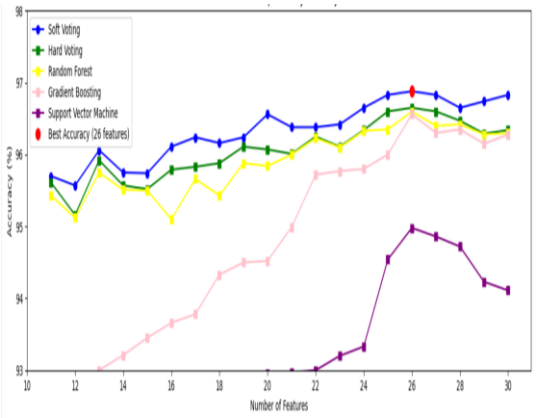


Figure 6. Feature Comparison

6. Conclusion

The proposed Phishing URL Detection System uses three classifier models such as Random Forest, Gradient Boosting, and Support Vector Machine to obtain better performance. The ensemble learning methods, hard voting and soft voting, are based on three models. The evaluation result of soft voting achieved the highest accuracy of 96.88%. According to the evaluation results of the ensemble methods, soft voting is better than hard voting.

In conclusion, the ensemble learning methods are an effective approach for detecting and preventing phishing attacks. The proposed system detects the reliable ensemble learning method of machine learning to attain better performance. Therefore, the system supports the users to avoid phishing attacks.

6.1. Limitation and Further Extension

In this paper, the system employs the UCI dataset, for training and testing data. Therefore, the system limits the new features that are not contained in the dataset. Moreover, the system only detects the URL; it does not detect voice phishing or text-based malicious message. The system takes a little computation time by using ensemble learning methods.

In future work, the system will collect and update the dataset for real phishing attacks. Furthermore, the system will block automatically malware URL found by adding browser extensions, such as Chrome, for more safety to users.

References

[1] Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf and Imtiaz Khan, “Phishing Attacks: A Recent Comprehensive Study and a New Anatomy”, Volume 3, Article 563060 March 2021

[2] Altyeb Taha, “Intelligent Ensemble Learning Approach for Phishing Website Detection Based on Weighted Soft Voting”, MDPI, Technology (IRJET) e-ISSN: 2395-0056, 4 November 2021

[3] Cing Gel Vung, Yu Yu Win, “URL Classification Based on Lexical Features by Machine Learning”, ICCA 2021 Date: 2021-02-25

[4] Mithilesh Kumar Pandey, Rekha Pal, Saurabh Pal, Arvind Kumar Shukla, Manish Ranjan Pandey, Shantanu Shahi, “Phishing Detection using Base Classifier and Ensemble Technique”, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169 Volume: 11 Issue: 11s, October 2023

[5] History of Phishing, https://www.phishing.org/history-of-phishing

[6] HAVOC SHIELD, “4 Phishing Steps that Cyber Criminals Use Against You”, July 12, 2023, https://blog.havocshield.com/en-us/4-phishing-steps-that-cyber-criminals-use-against-you

[7] Jason Brownlee, How to Perform Feature Selection with Numerical Input Data, https://machinelearningmastery.com/feature-selectio n-with-numerical-input-data, 18 August 2020

[8] Rami M. Mohammad, Fadi Thabtah, Lee McCluskey, “Phishing Websites Features”, 14 July 2015

[9] Batta Mahesh, “Machine Learning Algorithms - A Review”, (IJSR), ISSN: 2319-7064, ResearchGate Impact Factor (2018)

[10] Machine Learning, https://www.geeksforgeeks.org

[11] T. Vyvaswini1, Mr. P. P Nagaraja Rao, B. Kousalya, G. Pallavi, S. Abdullal5, P. Siddartha, “Phishing Website Detection using Machine Learning”, (IJARSCT), Volume 3, Issue 1, February 2023

[12] UCI Dataset, https://archive.ics.uci.edu/dataset/327/phishing+websites

[13] Kaggle Dataset, https://www.kaggle.com/datasets/akshaya1508/phishing-websites-detection? Select =Detection.xlsx

Classification of DDoS Attacks using Random Forest Algorithm

Su Su Naing, Khaing Khaing Wai
University of Computer Studies, Yangon
susunaing1@ucsy.edu.mm, khaingkhaingwai@ucsy.edu.mm

Abstract

A Distributed Denial-of-Service (DDoS) attack is a malicious cyberattack that overwhelms a target system with excessive traffic, rendering it inaccessible to legitimate users. This disruption can cause significant economic damage to businesses, enable extortion attempts, and even serve as a tool in cyber warfare. Therefore, DDoS attack classification system becomes important role for prevention system. This system can be utilized to distinguish both binary and multi classification on thirteen classes of CICDDoS2019 dataset and applied Random Forest (RF) Algorithm for both feature selection and classification. Evaluated models on both control features and selected features, then provided a comparative result with Extreme Gradient Boosting (XGBoost) and Gaussian Naïve Bayes (GNB). RF achieved the best performance, with 99.99% accuracy in binary classification using control features and 99.93% accuracy in multi classification using best selected features.

Keywords: Attack *Classification, CICDDoS2019 Dataset, Feature Selection, RF, XGBoost, GNB*

1. Introduction

DDoS attacks pose a significant cybersecurity threat, disrupting services and causing financial losses. Traditional detection methods often struggle to accurately identify DDoS attacks, leading to false positives and negatives. This research explores the use of RF algorithm to improve DDoS attack classification accuracy in both binary and multi-class scenarios.

2. Related Works

DDoS attacks are becoming increasingly developed and frequent, causing significant financial and infrastructural damage. As a result, a variety of many techniques for classification of DDoS attacks are proposed: In order to classify eleven DDoS attacks categories Decision Tree (DT), Support Vector Machine (SVM), Long Shor Term Memory (LSTM), and RF classifier was employed and analyzed a comparative result in [6] utilizing the CICDDoS2019 dataset. To enhance computational efficiency selected the top 20 most informative features using Extra Trees Classifier. RF model exhibited the highest overall accuracy among the evaluated algorithms.

The approach in [2] measured the performance of binary classification on Logistic Regression (LR), DT, RF, AdaBoost, K Nearest Neighbor (KNN), and Naive Bayes (NB) in detecting on each attack types using CICDDoS2019. Results indicated that LR, AdaBoost, and NB consistently outperformed than other models.

A comparative analysis of Gradient Boosting (GB), AdaBoost, and CatBoost algorithms for DDoS attacks detection was conducted in [1]. The study employed the CICDDoS2019 dataset, focusing on four exploitation-based attack types. To enhance model performance Extra Trees Classifier was applied to identify 25 most relevant features. Both full feature set and the reduced feature set were evaluated for both binary and multi-class classification scenarios.

3. Background Theory

The classification system for DDoS attacks plays a pivotal role in enhancing cybersecurity measures. By analyzing network traffic data found in the CICDDoS2019 dataset, the system can identify patterns and features unique to various types of DDoS attacks. This classification allows to categorize between benign and malicious traffic in specific type of attack, enabling more accurate detection of potential threats and also tends to help in the advancement of more robust intrusion detection systems.

3.1. Random Forest Classifier

Random Forest is a powerful machine learning algorithm that uses ensemble learning to create robust predictors. It constructs multiple decision trees, each trained on a random subset of data and features, reducing overfitting and improving generalization. To build each tree, information gain is used to optimize node splitting. Predictions

are made by aggregating the results of all trees through voting (classification) or averaging (regression).

3.1.1 Random Forest Algorithm

Let D = $\{(x_1, y_1),\ldots(x_N, y_N)\}$ represents the training data with $x_i = (x_{i,1},\ldots, x_{i,p})^T$.
For j = 1 to J:

1. Take a bootstrapped sample D_j from D that has size N.
2. Using the bootstrap sample D_j as the training data, fit a tree using binary recursive partitioning:
   a. Begin by grouping all of the observations into one node.
   b. Continue in a recursive manner until the stopping criterion is met for every unsplit node:
      i. From the p available predictors, choose m at random.
      ii. Using the m predictors from step i, identify the optimal binary split out of all the binary splits.
      iii. Use the split from step ii to divide the node into two descendant nodes.

To predict at a new point x,

$$\hat{f}(x) = \frac{1}{J}\sum_{j=1}^{J} \hat{h}_j(x) \text{ for regression} \quad (1)$$

$$\hat{f}(x) = \text{argmax}_y \sum_{j=1}^{J} I(\hat{h}j(x) = y) \text{ for classification} \quad (2)$$

Where, $\hat{h}_j(x)$ is the prediction of the response variable at x using the j^{th} tree

3.2. Information Gain

Information gain (IG) is a measure used to determine which feature should be used to split the data at each internal node of the decision tree. The higher IG, the better feature is for splitting the dataset.

IG = (Gini Index before the split)- (Gini Index after the split) (3)

3.3. Gini Index

The Gini Index is a statistical measure used to evaluate the impurity of a dataset. It ranges from 0 (perfect purity) to 1 (maximum impurity), with lower values indicating a more homogeneous dataset. Mathematically the Gini Index is represented by:

Gini Index = 1- ∑(Pi) No. of classes (4)

Where, (Pi) represents probability of i^{th} class
∑ is performed over all classes present in the dataset

3.4. Feature Selection

One of the significant advantages of Random Forest is its ability to assess the relative importance of features within a dataset. This is achieved through the Gini Importance metric, which measures the reduction in impurity (or uncertainty) when a particular feature is used to split a node in a decision tree. A higher Gini Importance value indicates that the feature is more influential in making accurate predictions.

4. Proposed System

The sophistication and frequency of DDoS attacks have surged, necessitating robust defense mechanisms. Accurate classification of DDoS attacks is paramount for effective prevention system. The researcher proposed binary and multi classification using both control (78 features) and selected (25 features) datasets. Consisting of a huge amount of up to date realistic network traffic, CICDDoS2019, is applied. Random Forest Algorithm is utilized for both distinguishing and selecting features to propose this model design and illustrated comparative results with XGBoost and GNB.

4.1. Dataset Description

The CICDDoS2019 dataset is a comprehensive collection of network traffic data encompassing both benign and malicious DDoS attack traffic flows. The dataset included 12 distinct modern DDoS attack types NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, TFTP and Benign. In that 57,518 benign samples and millions of malicious samples was involved. CICFlowMeterV3, a flow-based feature extractor, used to extract relevant network traffic features from pcap file.

4.2. Data Preprocessing

The data was cleaned, scaled and encoded before training on classifiers. This involved removing irrelevant 9 features Unnamed 0, Flow ID, Source IP, Destination IP, Source Port, Destination Port, Fwd Header Length.1, Inbound and SimilarHTTP of 88 features. Next, samples with missing values were substituted with the mean value of each column and infinite value was replaced with defined maximum value. Quantile Transformer is used to avoid effect of the outliers, handle non-normal distribution and reduce the training time.

4.3. System Flowchart

To perform the classification, imported and preprocessed the CICDDoS2019 dataset which was tested for both control and selected features for both binary and multi classification. 80% of data is used for training and 20% of data for testing. The system performance was evaluated on Accuracy, Precision, Recall, F1 score, and Confusion Matrix.

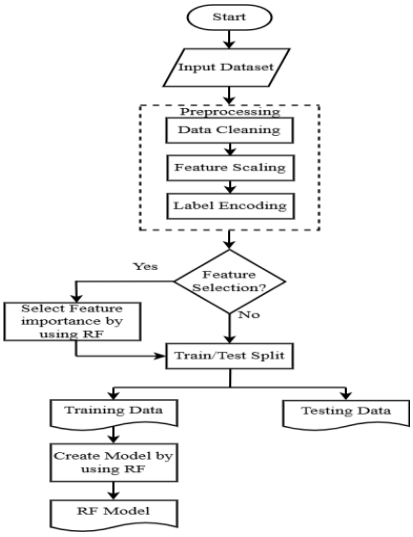


Figure .1 System flowchart for training

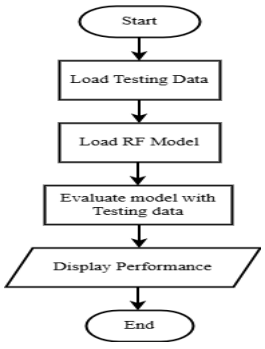


Figure .2 System flowchart for testing

Table 1. Comparison of DDoS attack classification works

Dataset	Classifier	Features	Multi-Classes	Performance (Accuracy)
CICDDoS2019 [6]	SVM, DT, LSTM, RF	20	Yes (11 classes)	99.32%
CICDDoS2019 [2]	LR, DT, RF, AdBoost, KNN, and NB	81	No	99.98%
CICDDoS2019 [1]	GB, AdBoost, CatBoost	78	Yes (4 classes)	97%
CICDDoS2019 (Random Forest System)	RF	78	Yes (13 classes)	99.99% Binary 99.89% Multi
		25		99.99% Binary 99.92% Multi

5. System Setup

To avoid discrepancies of model execution time, tested on one machine of Intel(R) Core (TM) i7-6700HQ CPU @ 2.60GHz 2.60 GHz, 8 GB RAM. This study implemented the models using Python programming language. Due to the dataset's massive size (over 70 million samples), a random subset of 300K samples was selected for each specific attack type to ensure reasonable training times and mitigate memory constraints on standard desktop computers. Additionally, the entire instances of benign and WebDDoS samples was utilized.

6. Experimental Results and Performance Evaluation

In this section, the result of control features and selected features are compared for Scenario A and Scenario B. Performance of all experiments are conducted with Accuracy, Confusion Matrix, F1 score, Precision and Recall. To compute above matrix, it is necessary to calculate:

TP: DDoS attack correctly identified as DDoS attack.

TN: Normal traffic correctly identified as normal.

FP: Normal traffic incorrectly identified as DDoS attack.

FN: DDoS attack incorrectly identified as normal.

Accuracy: The proportion of correct predictions out of total predictions.

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

Precision: The proposition of predicted trues are actually true.

$$\text{PRECISION} = \frac{\text{TP}}{\text{TP+FP}} \quad (6)$$

Recall: The proposition of actually trues are predicted as true.

$$\text{RECALL} = \frac{\text{TP}}{\text{TP+FN}} \quad (7)$$

F1-score: Harmonic mean of precision and recall.

$$\text{F1 Score} = \frac{2*\text{PRECISION}*\text{RECALL}}{\text{PRECISION}*\text{RECALL}} \quad (8)$$

Confusion matrix is a plot that visually summarizes how well a model performed on a dataset.

Training time helps in evaluating model efficiency and effectiveness in practical scenarios.

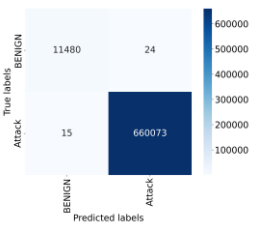
6.1. Binary Classification with Control Features

This section performed well on control result of RF and XGBoost as shown in Table .2. To compare these two algorithms, XGBoost was more efficient in terms of training time but RF generated more precise result with lowest 39 misclassified instances while XGBoost has 61 misclassified instances as shown in Fig .3. GNB, though faster, exhibited lower overall performance.

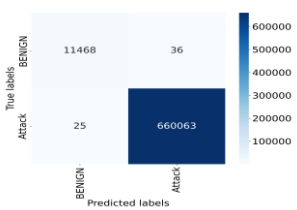
Table 2. Performance evaluation for binary classification with control features

Classifiers/ Performance	RF	XGBoost	GNB
Accuracy	99.99%	99.99%	96.12%
Precision	100%	100%	99%
Recall	100%	100%	96%
F1 score	100%	100%	97%
Training Time	1.94 min	0.23 min	0.15 min

RF



XGBoost



GNB

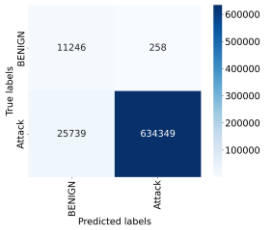


Figure .3 Confusion matrix

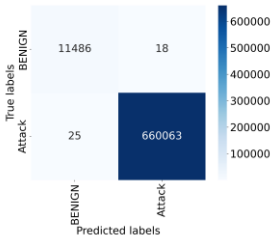
6.2. Binary Classification with Selected Features

In this session, to compare with binary classification with Control features, all of the algorithms were slightly decreased in training time and slightly improved accuracy of GNB but decreased in RF and XGBoost as displayed in Table .3. Similarly, misclassified instances of RF and XGBoost has increased about 43 and 78 as shown in Fig .4.

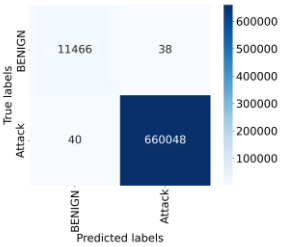
Table .3 Performance evaluation for binary classification with selected features

Classifiers/ Performance	RF	XGBoost	GNB
Accuracy	99.99%	99.98%	97.64%
Precision	100%	100%	99%
Recall	100%	100%	98%
F1 score	100%	100%	97%
Training Time	1.72 min	0.09 min	0.03 min

RF



XGBoost



GNB

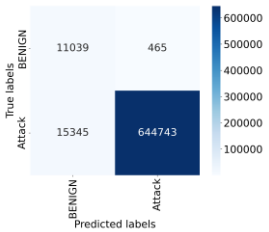


Fig .4 Confusion matrix

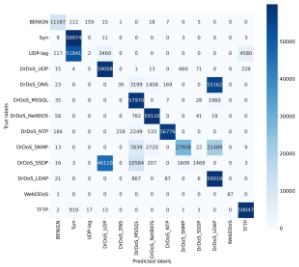


Figure .5 Confusion matrix

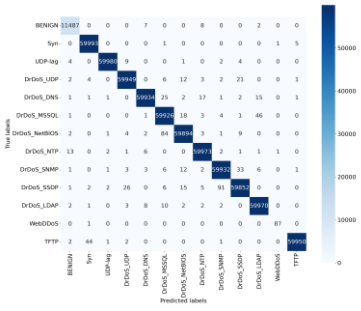
6.3. Multi Classification with Control Features

In this scenario, RF and XGBoost generally performed well in multiclass classification as displayed in Table .4. As shown in Fig .5, RF has the lowest misclassified instances about 665 while 2086 of XGBoost. GNB, on the other hand, exhibited significant misclassification issues, especially for UDPlag, DrDoS_DNS and DrDoS_SSDP.

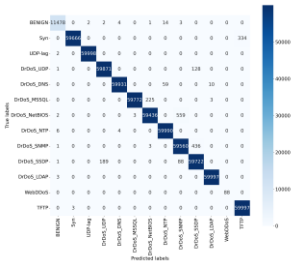
Table .4 Performance evaluation for multi classification with control features

Classifiers/ Performance	RF	XGBoost	GNB
Accuracy	99.89%	99.68%	67.24%
Precision	100%	100%	99%
Recall	100%	100%	98%
F1 score	100%	100%	98%
Training Time	4.07 min	1.55 min	0.15 min

RF



XGBoost



GNB

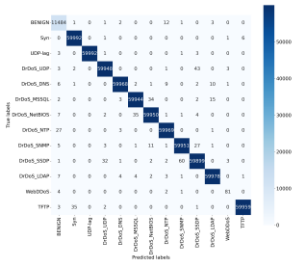
6.4. Multi Classification with 25 Selected Features

In this experimentation, RF and GNB are readily improved on performance than multiclassification with control features but XGBoost is slightly decreased as depicted in Table .5. RF has the lowest misclassified instances about 477 while 2340 of XGBoost as shown in Fig .6.

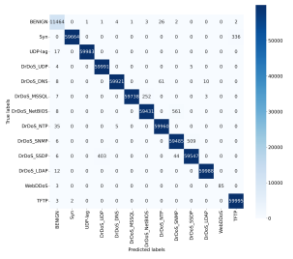
Table .5 Performance evaluation for multi classification with selected features

Classifiers/ Performance	RF	XGBoost	GNB
Accuracy	99.92%	99.65%	80.58%
Precision	100%	100%	84%
Recall	100%	100%	81%
F1 score	100%	100%	77%
Training Time	3.74 min	1.14 min	0.05 min

RF



XGBoost



GNB

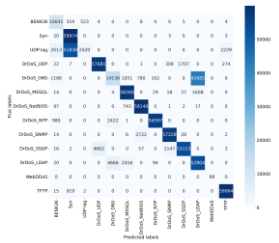


Figure. 6 Confusion matrix

6.5. Model Comparison

Overall RF achieved a better performance on all of the Sections. Specially in RF, binary classification with control feature is effective than the selected features as shown in Fig .7 and Fig .8 and multi classification offered a modest enhancement with selected features as illustrated in Fig .9 and Fig .10.

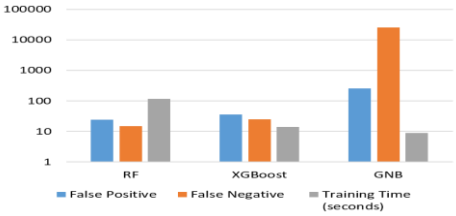


Figure .7 Binary classification with control features

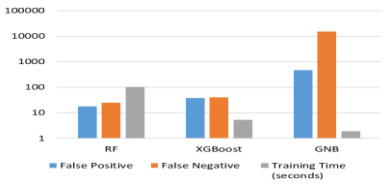


Figure .8 Binary classification with selected features

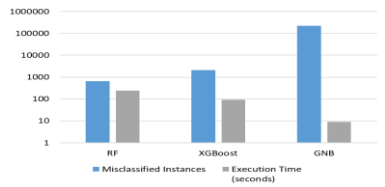


Figure .9 Multi classification with control features

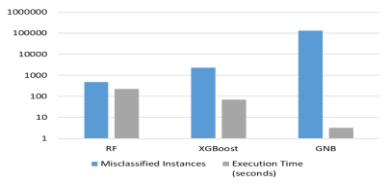


Figure .10 Multi classification with selected features

7. Conclusion

In recent years DDoS attacks are increasing as the demand for Internet connectivity massively grows. Thus, effective DDoS attack types classification system is quite important for preventing variety of DDoS attack. Investigated the classification of DDoS attack types using RF algorithms on larger and realistic datasets of CICDDoS 2019. This system will be deployed for future intrusion detection system in order to prevent DDoS attacks.

8. Limitations and Future Works

To reduce computational resources, the dataset was randomly selected 300K for each attack types except original smaller amount of benign and WebDDos. Despite the dataset's imbalance, the Random Forest model achieved high accuracy. To further enhance performance, balancing the dataset and exploring other machine learning and deep learning algorithms is recommended. Hybrid methods leverage the strengths of both approaches, enabling more precise and reliable classification of DDoS attacks. Additionally, investigating adaptive thresholding techniques that can dynamically adjust to changing network conditions and attack patterns can further improve the accuracy and effectiveness of DDoS attack detection systems.

References

[1] Denis Parfenov, Larisa Kuznetsova, Natalia Yanishevskaya, Irina Bolodurina, Arthur Zhigalov, Leonid Legashev, Research Application of Ensemble Machine Learning Methods to the Problem of Multiclass Classification of DDoS Attacks Identification, 2020

[2] Kishore Babu Dasari1, Nagaraju Devarakonda2, Detection of Different DDoS Attacks Using Machine Learning Classification Algorithms, 2021

[3] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining

[4] R. Al-Saadi, G. Armitage, J. But, and P. Branch, "A survey of delay based and hybrid TCP congestion control algorithms" IEEE Commun. Surveys Tuts., vol. 21, no. 4, pp. 3609-3638, 4th Quart., 2019

[5] Soumendu Chatteree, UNDERSTANDING FEATURE IMPORTANCE, 2022

[6] Yafei Xie, Machine learning-based DDoS detection for IoT networks, 2023

Facial Expression Classification by Using Local Binary Pattern And VGG16 Network

Aye Thant Thant Moe, Khant Kyawt Kyawt Theint
University of Computer Studies, Yangon
attmoe17@gmail.com, khantkyawtkyawttheint@ucsy.edu.mm

Abstract

Facial expression classification is a critical field of study in computer vision and artificial intelligence, with application from human-computer interaction. In this paper, using Local Binary Pattern (LBP) to extract features and VGG16 network's convolutional layer to extract features and then combining feature vectors as input to the convolutional neural network model to classify for facial expression classification. This system works in three parts, including preprocessing, feature extraction, and classifier of the image. The image preprocessing includes resizing, and normalization. For feature extraction, LBP is used to extract local features, and the VGG16 network is used to extract global features from the images. Lastly, the model is used to classify five emotions such as happiness, sadness, anger, disgust, and surprise. The experimental result of the proposed model was tested on Extended Cohn Kanade (CK) dataset and obtained a 98% accuracy rate.

Keywords: *Image Classification, Local Binary Pattern, VGG16 Network*

1. Introduction

Facial expression is one of the most important ways for humans to communicate their emotional states in daily life. Facial expression classification is the most used of study in computer vision and artificial intelligence, with applications including, human computer interaction, healthcare, security, marketing and entertainment and so on. In the 1970s, Ekman and Friesen identified five essential emotions: happiness, surprise, sadness, fear, disgust, and anger [7]. Each emotion has its own form. Thus, the training data set should be labeled. The classifier detects the input images after being trained by assigning a unique class label to it.

In this system, LBP is a texture descriptor that captures the local structure of an image by comparing each pixel with its neighboring pixels, resulting in a binary pattern that represents the image's texture. On the other hand, VGG16 networks are deep learning models that automatically learn hierarchical features from raw image data. By integrating a VGG16 with LBP features, the resulting features are used as input to a classification network, which predicts each emotion class of facial expression. This model not only enhances the accuracy of the classification but also provides a more comprehensive understanding of the facial expressions by utilizing both texture and deep features. The results from this study could contribute to more accurate and reliable facial expression classification system.

In this paper, section 1 describes an introduction, section 2 related works, section 3 background theory, section 4 proposed system, section 5 datasets description, section 6 performance evaluation, and section 7 conclusion of the proposed system.

2. Related Works

Facial expression classification has received a lot of attention in recent years, thanks to advances in machine learning and deep learning, which have used multiple models and methodologies to enhance accuracy and performance results. There are a variety of techniques for the classification of facial images. [1] The model used to classify machine learning algorithms. The features' dimensionality was reduced using principal component analysis (PCA), and then support vector machines (SVM) classified selected features into facial emotions. The model was trained on the Extended Cohn-Kanade Database (CK+) and then achieved a recognition rate of 92.63%. [2] The model was evaluated by

integrating the Histogram of Oriented Gradient (HOG) with the convolutional neural network (CNN) model. HOG is estimated from each image in the dataset. The resulting dataset is applied to a convolutional neural network (CNN). The model was tested on the CK+ dataset and obtained an accuracy rate of 96.73%. Therefore, the system will be able to achieve accurately recognition of emotions used to learn deep learning algorithms.

3. Background Theory

The facial expression classification system is the most used in deep learning methods, especially, convolutional neural network (CNN) to improve classify face expression. CNN is the most used in object detection and image classification. CNN consists of multiple layers and neurons in each layer are directly connected with the neurons in the next layer that are relevant for the task. CNN consists of input layer, hidden layer, and output layer. Input layer includes convolutional layer, pooling layer to reduce the dimension of the feature maps before sent to data to the network. The output layer also called fully connected layer includes dense layers and utilizes softmax activation function to compute probabilities of each class.

4. Proposed System

The architecture of the proposed system consists of three processes, including preprocessing of the image, extraction of features, and classification of a facial expression. The first step is image preprocessing for improving the quality of image. Secondly, the adaptation of feature extraction using LBP and VGG16 approaches of an input image, and lastly, combined feature vectors connected convolutional neural network is applied to classify facial expressions, including happiness, sadness, disgust, anger, and surprise.

In this system, the face dataset is used from the Kaggle and classified multiclass emotion expression. The dataset is divided into 80% of data for training and 20% of data for testing. Figure .1 depicts overview of the proposed system.

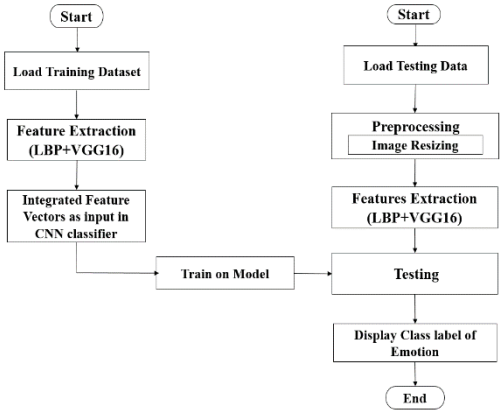


Figure .1 Overview of the proposed system

4.1 Image Preprocessing

Image preprocessing is a critical step in preparing raw image data for analysis by machine learning and deep learning models. It involves various techniques to enhance image quality, normalize features, and reduce computational complexity. Common preprocessing steps include resizing images to a uniform size, converting images to grayscale, and normalizing pixel values to a specific range. These processes help standardize the input data, making it more suitable for feature extraction and improving the overall performance of image classification and recognition tasks.

For an grayscale image represented as I(x,y), where x,y are spatial coordinates, and normalization maps the pixel intensity I(x,y) from its original range [0,255] to a normalized range [0,1]:

$$I_{norm(x,y)} = \frac{I(x,y)}{255} \quad (1)$$

Where,

I(x,y) = the grayscale intensity at pixel (x,y)

$I_{norm(x,y)}$ = the normalized pixel value.

4.2 Feature Extraction

The design of algorithms for extracting information on face expression from the image that has been pre-processed is the second phase in facial expression classification approach. In this system, there are used two approaches for feature extraction. The first approach is LBP, and the second is the convolutional layer of the VGG16 network to extract features at the same time and then combine the feature vectors as input to the convolutional neural network model. Face

localizing and its features as in an image is the method used to extract data about the facial expression from a static image. The main facial features are eyebrows, eyes, nose, mouth and chin.

4.2.1 Local Binary Pattern

LBP is the most widely used texture descriptor for face classification in computer vision and image processing. LBP is the type of appearance-based features. It extracts the local texture features of an image. A facial picture is segmented into small local regions, which are then utilized to extract LBP feature distributions and concatenated into an improved feature vector to serve as a face image descriptor. The LBP method uses a grayscale image, which means it does not capture color image information in the texture pattern. It was first described in 1994[4], There are many ways to generate LBP code for every selected pixel of the input image.

- Central LBP
- Mean LBP
- Median-LBP
- Gradient-LBP
- Palladian-LBP

Among them, the Central LBP approach is used to extract features in this system. It operates by comparing the intensity of the central pixel value and the intensity of its eight neighboring pixels in the small neighborhood pixel. The binary code for each pixel in the neighborhood is set depending on whether its intensity is greater than or less than the intensity of the central pixel. These binary values are generated by generating clockwise rotation as a binary number, which represents the texture of that neighborhood. As a resulting feature vector, these binary numbers are constructed as a histogram of the texture distribution within an image. Mathematical equation of the LBP code of a pixel (x_c, y_c) is given by:

$$\text{LBP}\,(x_c, y_c) = \sum_{p=0}^{7} s(g_p - g_c)2^p \qquad (2)$$

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

Where,

P = total number of sample points of the pixel

g_c = the intensity value of the central pixel

g_p = the intensity value of its neighbors pixel

4.2.2 Visual Geometry Group (VGG16)

VGG16 is a network in deep convolutional neural network architecture that extracts hierarchical features from input images. It was developed by the Visual Geometry Group at the University of Oxford in 2014[8]. This network consists of 16 weight layers, including 13 layers and 3 fully connected layers with ReLU activation applied after each convolution. The convolutional layers use small 3x3 filters with a stride of 1 and padding to preserve spatial dimensions, enabling fine grained feature extraction. Pooling layers, typically max-pooling with a 2x2 window and a stride of 2, reduce spatial dimensions progressively, capturing larger information while maintain computational efficiency. In the initial layers, VGG16 extracts features such as edges, textures, and corners including object parts and spatial hierarchies relevant to facial expressions for facial expression classification.

4.3 Classification

The face dataset is classified using convolutional neural network (CNN). CNN consists of input layer, hidden layer, and output layer. In this system, the integrated feature vectors into input layer of CNN and used the Adam optimizer to train, added dense layer, and output layer of CNN is used Softmax function to classify probabilities of class. The dataset is divided into 80% of data for training and 20% of data for testing. The performance of the system was determined on Confusion Matrix, Accuracy, Precision, Recall, and F1-score. The CNN architecture is shown in the following figure 2.

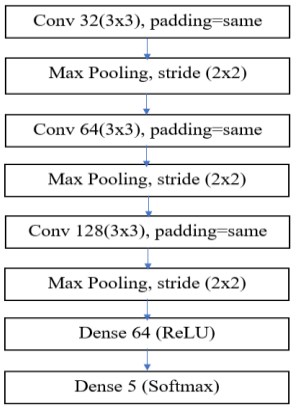


Figure 2. The CNN architecture

The convolutional layers use small 3x3 filters with a stride of 1 and padding to preserve spatial

dimensions to feature extraction. The max-pooling layer with a 2x2 window and a stride of 2 is utilized to reduce spatial dimensions. After convolution layers, the resulting feature vector as input into output layer also called dense layer of model, which are fed into a softmax layer for classification.

5. Datasets Description

The three datasets are used for the implementation of the facial expression classification system.

5.1. Extended Chon Kanade (CK) Dataset

The Extended Cohn Kanade (CK) dataset is a public test dataset for facial expression classification system. This dataset consists of 852 grayscale face images with a size of 48 x 48 pixels. This dataset contains five different classes of facial expression, including anger, happiness, sadness, disgust, and surprise. Figure 3 shows the sample of face images.

Figure 3. The sample images of Extended Chon Kanade (CK) dataset

5.2. Unknown Dataset

The dataset contains 943grayscale face images with the size of 48x 48 pixel. It is type of PNG format file. It has been label for each class, including anger, happiness, sadness, disgust, and surprise. Figure 6 shown the sample of images.

Figure 4. The sample images of Unknown dataset

5.3. FER2013 Dataset

The dataset contains 13645 grayscale face images with the size of 224 x 224 pixel. It is type of PNG format file. It has been label for each class, including anger, happiness, sadness, disgust and surprise. It has the various position of face such as different lighting condition, posed faces, different age groups, and cartoon images. So, it is complexity and diversity the faces. Figure 7 shows the sample of face images.

Figure 5. The sample images of FER2013 dataset

6. Performance Evaluation

The proposed model was trained and tested on Extended Cohn Kanade (CK), Unknown, and FER2013 dataset. Confusion matrix, Precision, Recall, F1-score and Accuracy are used to measure for testing the classification methods of expressions and determine the system's efficiency. To compute above matrix, it is necessary to calculate:

TP: the model correctly predicts the positive emotion

TN: the model correctly predicts the negative emotion

FP: the model incorrectly predicts the positive emotion when it's actually negative.

FN: the model incorrectly predicts the negative emotion when it's actually positive.

Accuracy: the proportion of accurate predictions to all predictions.

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

Precision: the ratio of predicted positive value from all the positive values.

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

Recall: the portion of relationship between predicted positive value and all predicted value.

$$\text{Recall} = \frac{TP}{TP+FP} \quad (5)$$

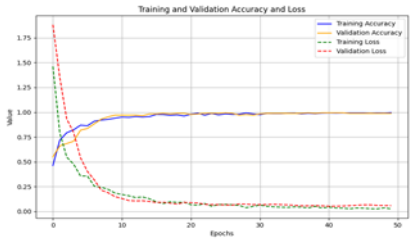
F1-score: The harmonic mean of precision and recall.

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

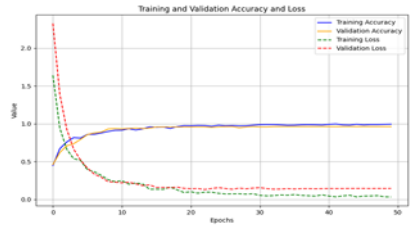
Confusion matrix displays a square matrix that describe the number of data examples known to be in true label group and predicted label group.

6.1. Experimental Results of Proposed Model

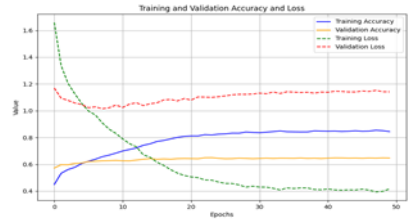
The dataset Extended Cohn Kanade (CK), Unknown, and FER2013 datasets were separately input into the proposed model for training. And then, the classification rate of Extended Cohn Kanade (CK) dataset is obtained 99%, Unknown dataset is obtained 96%, and FER2013 dataset is obtained 65% accuracy rate of respectively.



(a)



(b)



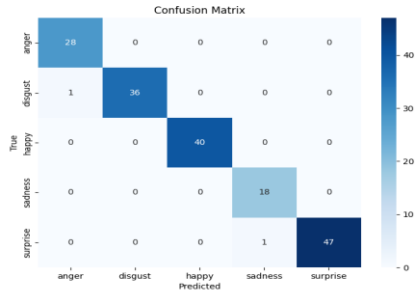
(c)

Figure 6. Plotting train_acc and val_acc of the proposed model (a) Extended Cohn Kanade (CK) dataset, (b) Unknown dataset, and (c) FER2013 dataset during epoch 50

6.2 Performance Evaluation of Extended Cohn Kanade (CK) dataset

	precision	recall	f1-score	support
anger	0.97	1.00	0.98	28
disgust	1.00	0.97	0.99	37
happy	1.00	1.00	1.00	40
sadness	0.95	1.00	0.97	18
surprise	1.00	0.98	0.99	48
accuracy			0.99	171
macro avg	0.98	0.99	0.99	171
weighted avg	0.99	0.99	0.99	171

(a)



(b)

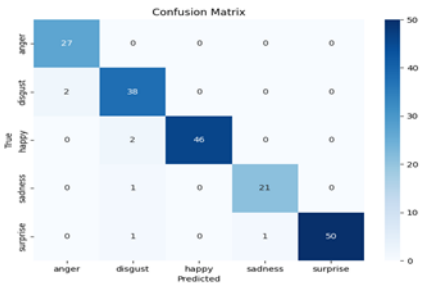
Figure 7. (a) Classification report and (b) confusion matrix of Extended Cohn Kanade (CK) dataset

In this confusion matrix, when the Extended Cohn Kanade (CK) dataset is trained by proposed model, disgust, happy, and surprise reached 100% accuracy rate, anger reached 97% accuracy rate, and sadness reached 95% accuracy rate.

6.3 Performance Evaluation of Unknown dataset

	precision	recall	f1-score	support
anger	0.93	1.00	0.96	27
disgust	0.90	0.95	0.93	40
happy	1.00	0.96	0.98	48
sadness	0.95	0.95	0.95	22
surprise	1.00	0.96	0.98	52
accuracy			0.96	189
macro avg	0.96	0.96	0.96	189
weighted avg	0.96	0.96	0.96	189

(a)



(b)

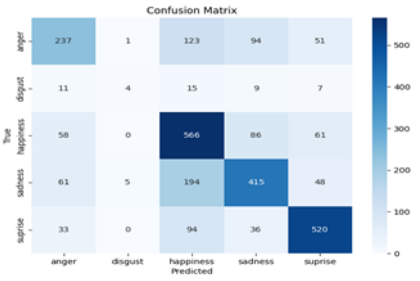
Figure 8. (a) Classification report and (b) confusion matrix of Unknown dataset

In this confusion matrix, when the Unknown dataset is trained by proposed model, happy, and surprise reached 100% accuracy rate, sadness reached 95% accuracy rate, anger reached 93% accuracy rate and disgust reached 90% accuracy rate.

6.4 Performance Evaluation of FER2013 dataset

	precision	recall	f1-score	support
anger	0.60	0.52	0.56	506
disgust	0.45	0.20	0.27	46
happiness	0.60	0.69	0.64	771
sadness	0.63	0.59	0.61	723
suprise	0.76	0.78	0.77	683
accuracy			0.65	2729
macro avg	0.61	0.56	0.57	2729
weighted avg	0.65	0.65	0.64	2729

(a)



(b)

Figure 9. (a) Classification report and (b) confusion matrix of FER2013 dataset

In this confusion matrix, when the FER2013 dataset is trained by proposed model, surprise is obtained accuracy rate of 76%, sadness is obtained accuracy rate of 63%, anger and happiness reached accuracy rate of 60%, and disgust reached accuracy rate of 45% respectively.

7. Conclusion

This paper integrates LBP and the convolutional layer of the VGG networks, which are used to extract features, and then the resulted features are fed to a convolutional neural network model for better facial expression classification. After running 5 times, the proposed model was tested on datasets, and it obtained 98%, 95%, and 65% average accuracy rates in the Extended Cohn Kanade (CK) dataset, Unknown dataset, and FER2013 dataset, respectively. For future works, the proposed model will be extended to optimize in real-time applications such as human-computer interaction, evaluation in the classroom, monitoring of healthcare, marketing, and behavioral psychology.

References

[1] Amornvit Vatcharaphrueksadee, Maleerat Maliyaem, Facial Emotion Classification of Multi-Type Datasets based on SVM Classifier, IEEE, 2022.

[2] Nadia Shamsulddin Abdulsattar, Mohammed Nasser Hussain, Facial expression recognition using HOG features with convolutional neural network, 2022.

[3] Yang Tao, Yuanzi He, Face Recognition Based on LBP Algorithm, ICCNEA, 2020.

[4] Benisha S, Mirnalinee TT, Human Facial Emotion Recognition using Deep Neural Networks, The International Arab Journal of Information Technology, Vol. 20, No. 3, May 2023.

[5] Koteswara Rao Kolli, Phaninder Vinay, Ramesh Boddu, K. Naresh, HOG BASED EMOTION RECOGNITION USING FACIAL IMAGE FEATURES, 2023.

[6] Mahdi Jampour, Malihe, Multiview Facial Expression Recognition, A Survey, IEEE, 2022.

[7] Ekman, P and Friesen, W.V, “Constant across cultures in the face and emotion”, Journal of personality and social psychology, 1971.

[8] Srikanth Tammina, Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images, International Journal of Scientific and Research Publications, Volume 9, Issue 10, October 2019.

Strengthening Web Security for Input Validation Approach in PHP Web Development using Knuth-Morris-Pratt Pattern Matching Algorithm

Su Sunndy Tun, Nilar Aye
University of Computer Studies, Yangon
susunndytun@ucsy.edu.mm, nilaraye@ucsy.edu.mm

Abstract

Web applications are widely used for various services, which makes them targets for cyberattacks. Among the common vulnerabilities of web applications are SQL Injection (SQLi) and Cross-Site Scripting (XSS) attacks that exploit weak input validation mechanisms. An input validation technique is proposed using the Knuth-Morris-Pratt (KMP) pattern-matching algorithm to mitigate these threats. The KMP algorithm efficiently detects malicious input patterns by preprocessing and avoiding redundant comparisons, ensuring real-time security enhancements. KMP is compared with existing input validation techniques to demonstrate its effectiveness in improving detection accuracy. Integrating PHP-based web development enables filtering malicious input, SQLi, and XSS from input forms. An algorithm checks for these attacks; if detected, they are rejected and logged. It offers a lightweight and efficient solution to prevent web vulnerabilities and valuable insights to web developers, researchers, and the cybersecurity community. Future enhancements may include interoperability with other programming environments and performance optimizations, marking a significant advancement in securing web applications.

1. Introduction

The widespread use of web applications in daily life has increased security concerns, particularly around input forms, which are common targets for attacks like SQLi and XSS [6].

The proposed system is implemented using the PHP programming language, one of the simplest and most powerful server-side scripting languages, many websites are developed in that language [15]. While PHP offers several benefits, applications built with it often face security issues due to poor user-input handling. However, traditional validation methods, such as regular expressions, are limited when facing sophisticated threats. A better and faster algorithmic approach is needed for the problem of input validation.

To address this, the proposed system integrates the **Knuth-Morris-Pratt (KMP) algorithm**, a highly efficient pattern-matching technique, to improve input validation and detect malicious patterns in user inputs [2]. This approach enhances PHP application security through a faster and more reliable detection mechanism.

The structure of this paper is as follows: Section 2 reviews related work, while Section 3 covers the system background, including web application vulnerabilities such as SQLi and XSS attacks and the KMP pattern-matching algorithm. Section 4 presents the proposed system, and Section 5 evaluates the results. Finally, Section 6 concludes the paper and outlines potential.

2. Related Work

The paper aims to improve web application security by reducing risks from attacks. One of the first studies in this area is "Web Application Vulnerability Testing System Using XSS_SQL_Scanning Algorithm," proposed by Thinzar Aung from the University of Computer Studies (UCSY) [5]. A Web Application Vulnerability Testing System detects SQLi or XSS vulnerabilities when a URL is entered into an input box using the XSS_SQL_Scanning algorithm.

In another study, the research paper titled “Detecting SQL Injection Attacks in Online Learning System Using Rabin Karp Pattern-matching Algorithm” was submitted by San San Wai from UCSY [3]. The primary focus is on utilizing the Rabin-Karp algorithm to detect SQLi in online learning platforms by applying hash-based pattern recognition.

Another study is "A novel technique to prevent SQL injection and Cross-Site Scripting attacks using Knuth-Morris-Pratt string match algorithm". Oluwakemi Christiana Abikoye, Abdullahi Abubakar, Ahmed Haruna Dokoro, Oluwatobi Noah Akande, and Aderonke Anthonia Kayode submitted this paper [2]. This study uses JavaScript Programming Language by integrating the KMP algorithm to prevent SQLi and XSS attacks.

These studies provide a foundation for this paper, which focuses on integrating KMP into

PHP development to strengthen input validation and improve web application resilience.

3. Background

3.1. Common Web Application Vulnerabilities

The OWASP Top 10 for 2024 identifies the most critical web application security vulnerabilities for developers and organizations to enhance their security practices [10]. There are:

1. Broken Access Control
2. Cryptographic Failures (Sensitive Data Exposure
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

3.2. Input Validation Techniques in PHP Web Development

Input validation is a critical aspect of web application security that ensures user data meets the required standards before processing [8]. Traditional methods like **regular expressions** and **PHP's built-in functions** (e.g., **preg_match ()**, **filter_var ()**) are commonly used but may suffer from inefficiency and inconsistency when dealing with large or complex pattern sets.

To address these limitations, the KMP algorithm offers a more performant alternative. Compared to regex, KMP demonstrates faster and more scalable performance, especially under high-load environments. Furthermore, KMP can be practically integrated into real-world PHP systems by embedding it into form validation processes. KMP in input validation enhances both detection accuracy and performance, making it a strong candidate for secure, efficient input handling in PHP applications [2].

3.3 Common Security Threats in PHP Web Development

PHP is powerful, but requires developers to address certain security issues to protect it from various threats. These threats arise due to poor input validation [11], inadequate data handling, or misconfiguration. Some of the common security threats in PHP web development [9]:

1. **SQL Injection (SQLi):** This attack manipulates input fields to inject malicious SQL queries, potentially exposing, altering, or deleting sensitive database data [1].
2. **Cross-Site Scripting (XSS):** XSS exploits poor input sanitization to inject harmful scripts into web pages, allowing attackers to steal data like cookies and session tokens, leading to unauthorized access [4].

4. Overview of the Proposed System

The development process begins with studying and collecting **known SQLi** and **XSS attack payloads** from trusted websites such as **OWASP** (Open Web Application Security Project), **PayloadsAllTheThings** (GitHub Repository), etc.

A **payload** refers to the malicious data or code injected into a system's input field with the intent of testing or exploiting a vulnerability. Analyzing these payload structures, behaviors, and real-world examples to generate custom attack pattern rules for detection.

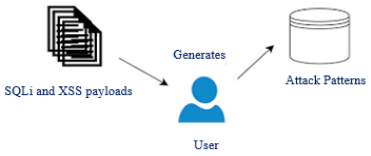


Figure 1. Generating attack patterns

Once the attack patterns are predefined, they are stored in a backend **database** for preprocessing. Here are some of the attack pattern rules:

Table 1. Attack Patterns

No	Pattern	Type
1	{"key":"<script>"}	XSS
2	data:text/html,<script> alert(1)</script>	XSS
3	*\x3cscript> alert(1) </script>	XSS
4	<	XSS
5	') OR '1'='1' --	SQL Injection
6	ALTER TABLE	SQL Injection
7	SELECT * FROM	SQL Injection

After that, inputs can be submitted through the system's **login** or **registration** forms. Users may enter inputs or payloads manually, one at a time, or use automated tools to reduce testing time. There are various automated tools widely used in the field of web security, and in this system, **OWASP ZAP (Zed Attack Proxy)** [12] (version 2.15.0) is utilized for testing.

ZAP is a widely used open-source security tool for simulating real-world attack scenarios by automatically submitting a large number of payloads within a minute. It supports **fuzzing** [12] through **file fuzzers**, which use curated payload libraries like **jbrotuzz** to automate and simulate real-world attack scenarios. Fuzz testing, or fuzzing, involves automatically submitting a wide range of payloads, both **SQLi** and various forms of **XSS** attack inputs, to the system.

All inputs submitted through the ZAP fuzz testing tool are matched against a set of **predefined attack-pattern rules** stored in the system's database. When a match is found, the system **immediately blocks the malicious input** and **notifies the administrator**, ensuring a real-time response to active threats.

Finally, the detection process is completed, the system logs all incoming payloads and categorizes them by attack type, allowing the administrator to view a visual representation on the dashboard. Additionally, all input data can be downloaded as a report for further analysis.

The system's detection accuracy is a key performance metric that reflects how effectively it identifies malicious inputs such as SQLi and XSS attacks. It is automatically calculated and displayed that is based on the system's ability to correctly classify each input as either malicious or legitimate.

The integration of KMP enables the system to perform **fast, accurate,** and **efficient pattern detection** without unnecessary comparisons. This design not only improves the **speed** and **reliability** of the detection process but also ensures the system remains scalable and effective under intensive testing conditions.

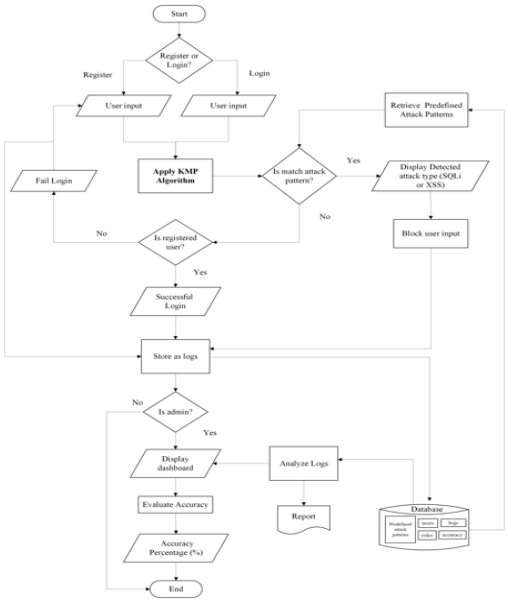


Figure 2. Workflow of Input Validation Approach in PHP Web Development Using Knuth-Morris-Pratt Pattern-Matching Algorithm

4.1 Proposed Algorithm

There are several pattern-matching algorithms, such as Naïve, Boyer-Moore, Rabin-Karp, and KMP, each with strengths and weaknesses. The KMP algorithm is one of the most efficient pattern-matching algorithms used in computer science to find specific sequences within large datasets, including text files, databases, and web applications [2]. The proposed algorithm detects SQLi and XSS attacks by efficiently matching input patterns. The implementation of this algorithm is presented in the Algorithm, and a detailed explanation of its working process is provided in the following subsection.

1. Preprocessing Phase

Algorithm Knuth-Morris-Pratt (KMP) algorithm

1: Input: The pattern to search for
2: Output: Array containing the lengths of the longest prefix that is also a suffix for each position in the pattern
Compute Prefix Function
3: m ← Length of the pattern
4: prefix_table ← Array of size m, initialized with all zeros
5: k ← 0 (Tracks the length of the current prefix-suffix match)
Iterate through the pattern starting from the second character (i = 1):
6: While k > 0 and pattern[k] ≠ pattern[i]:
7: Set k ← prefix_table[k - 1] (Use the prefix table to find the next possible match length)
8: If pattern[k] = pattern[i]:
9: Increment k
10: Assign prefix_table[i] ← k
11: Return prefix_table

1) Preprocessing Phase: e prefix table (also called the LPS table) is built to record the length of the longest prefix that is also a suffix for each position in the pattern. It starts by initializing the table with zeros and compares characters to identify repeated patterns. When a match is found, the value of k is stored, allowing the algorithm to skip redundant comparisons during the search.

2. Pattern-Matching Phase

Algorithm Knuth-Morris-Pratt (KMP) algorithm

1: Input:
text: The text to search in
pattern: The pattern to search for
2: Output: Starting indices of all occurrences of the pattern in the text Compute Match Function
3: n ← Length of the text
4: m ← Length of the pattern
5: prefix_table ← Call the prefix function on the pattern
6: q ← 0 (Tracks the number of characters currently matched)
Iterate through the text (i = 0 to n - 1):
7: While q > 0 and pattern[q] ≠ text[i]:
8: Set q ← prefix_table[q - 1] (Use the

prefix table to backtrack and find the next possible match length)
9: If pattern[q] = text[i]:
10: Increment q
11: If q = m (All characters of the pattern matched):
12: Print "Pattern found at index i – m + 1"
13: Set q ← prefix_table[q - 1] (Prepare to look for the next match)

2) Pattern-Matching Phase: The KMP algorithm is a fast way to find a pattern in a text by using a prefix table to skip unnecessary comparisons. It scans the text, keeping track of matched characters (q). If there's a mismatch, it uses the prefix table to jump to the next possible match instead of starting over by using the shifting formula:

$$\text{Shift} = \text{Current position of the pattern} - \text{Prefix Value at previous Character} \quad (1)$$

When a full match is found (q == m), the algorithm prints the starting index and continues searching. This smart shifting method allows KMP to work in linear time O(n + m), making it much faster than traditional search methods, especially for large texts.

4.2 Penetration Testing of the Target System

Penetration testing is an important part of web security that helps identify weaknesses by simulating real-world attacks on systems, networks, or applications. Within this process, fuzzing plays a key role by automatically sending a wide range of unexpected or malicious inputs to the system. [7].

4.3. Automated Fuzz Testing for SQLi and XSS

To evaluate the system's ability to detect SQLi and XSS attacks, a fuzz test was initially conducted using the ZAP tool, as illustrated in the figures.

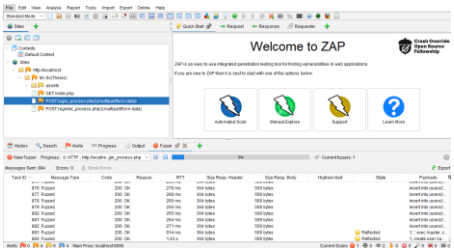


Figure 3. Fuzz Testing for SQLi Attacks

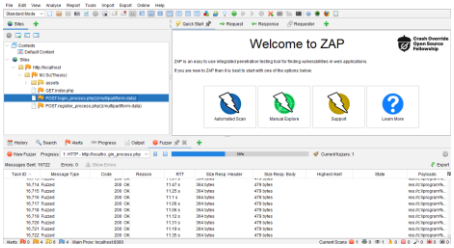


Figure 4. Fuzz Testing for XSS Attacks

During SQLi and XSS attack fuzzing, payloads are continuously submitted to the system's input form, and the process is considered **complete once 100%** of the payloads have been executed and processed by the system.

4.4. Log Analysis

Figure 5 displays all logs of detected inputs after fuzzing used by **ZAP,** categorized into four main categories: **SQL Injection**, **XSS Attacks**, **Legitimate Inputs, and Other Inputs** within a web security system. The administrator enables the export of logs for further review or forensic analysis and calculates the accuracy of each attack.

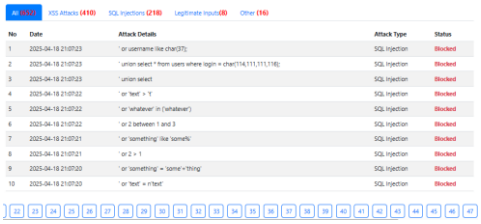


Figure 5. Logs Overview

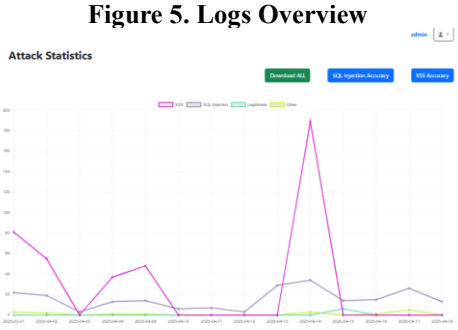


Figure 6. Graphical Representation of Logs Overview

4.5. Detection and Classification of User Input

In the proposed system, every user-submitted input is systematically analyzed using KMP. The following figures illustrate the classification of user input detections, offering a clear and structured visualization of how the proposed system identifies, categorizes, and responds to different types of inputs.

The figures below present the detection results of all inputs or payloads, displayed in graphical formats.

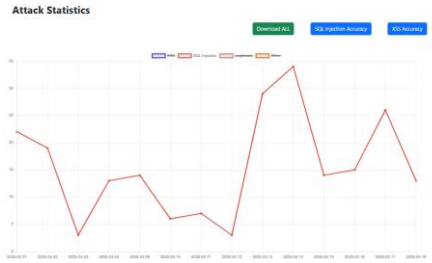


Figure 7. Detection of SQLi Attacks

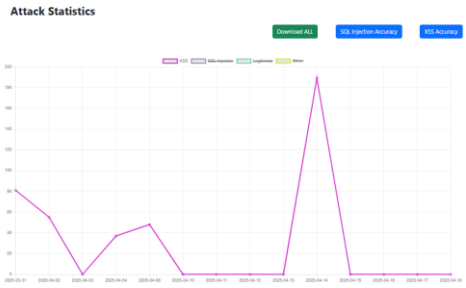


Figure 8. Detection of XSS Attacks

Figure 9. Logging in as a legitimate user

Figure 10. Detection of other

The system is capable of detecting a variety of SQLi and XSS attacks submitted through ZAP, including for SQLi: 1) Active SQL Injection, 2) MS SQL Injection i, 3) MS SQL Ninja Injection (Blind), 4) MySQL Injection (Blind), 5) MySQL Injection 101, 6) MySQL/MS SQL Common Injection, 7) Oracle SQL Injection, 8) Passive SQL Injection, 9) SQL Injection.

For XSS attacks: 1) URI Cross-Site Scripting, 2) URL Breaking, 3) XSS 101,4) XSS 102, 5) XSS Embed/Evade, 6) XSS Gecko, 7) XSS HTML Breaking, 8) XSS Image Tag, 9) XSS Internet Explorer, 10) XSS JS Breaks, 11) XSS Style Injection.

5. Experimental Result

In the experimental phase, the system was evaluated on its ability to detect and classify SQLi and XSS attacks correctly. For each type of attack, accuracy was computed by measuring the proportion of attack payloads correctly identified as malicious out of the total number of submitted payloads, which also included those misclassified.

The detection logic identifies and categorizes each submission into one of three types, as shown in Table 2:

Table 2. Specification of Detection Logic

Input	Description
SQLi or **XSS**	Correctly detected as an attack.
Other	Incorrectly classified or missed detections.
Legitimate:	Legitimate inputs are normal user data, not the focus of this evaluation.

This helps improve web security by showing how well the system works in real-time and evaluating the system's accuracy in detecting SQLi and XSS attacks effectively. To evaluate the effectiveness of the proposed system, accuracy tests were conducted separately for both. The detection accuracy is computed using the following equation:

Accuracy (%) = (Number of Attacks Detected / Total Payloads) * 100 (2)

5.1 Accuracy Rate for SQL Injection

The proposed system was tested using SQL Injection payloads from ZAP to measure its detection accuracy as described in Table 2.

Table 3. Accuracy Rate for SQL Injection

Detection	Value
Detected as SQLi Attack	28018
Detected as other	1416
Total Payloads	29434
Calculation	(28018/ 29434) *100
Accuracy Rate (%)	**95.19 %**

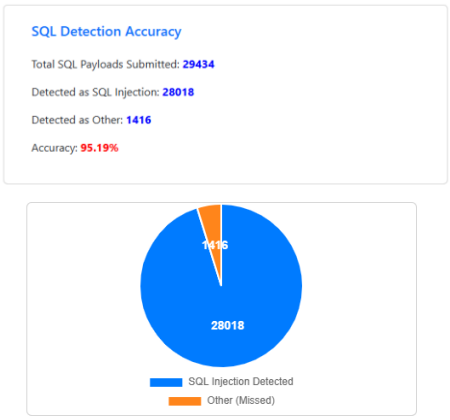


Figure 11. Accuracy Rate of SQLi Attack

5.2. Accuracy Rate for Cross-Site Scripting

The proposed system was evaluated for XSS attack detection using payloads from the ZAP tool.

Table 4. Accuracy Rate for Cross-Site Scripting

Detection	Value
Detected as XSS Attack	42016
Detected as other	624
Total Payloads	42640
Calculation	(42016/ 42640) *100
Accuracy Rate (%)	**98.54 %**

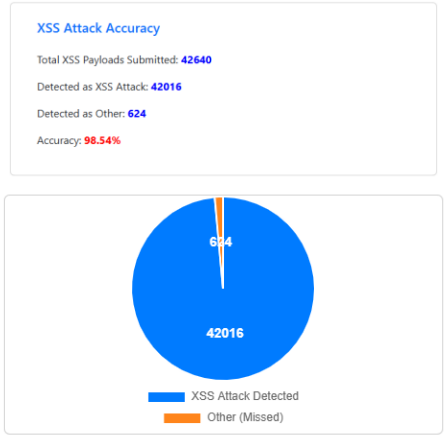


Figure 12. Accuracy Rate for XSS Attacks

6. Conclusion and Future Work

This study enhances web application security by integrating the Knuth-Morris-Pratt (KMP) algorithm into PHP-based input validation to detect common vulnerabilities such as SQL Injection and Cross-Site Scripting. By analyzing user inputs and comparing them with predefined malicious patterns, the system efficiently identifies and blocks attacks. Secure coding techniques like parameterized queries are combined with KMP's fast pattern-matching to support real-time application protection. The system also provides visual analytics and demonstrates strong detection accuracy, adaptability, and practical use for developers and researchers.

The KMP algorithm can detect attacks such as Command Injection, Path Traversal, Remote File Inclusion, Local File Inclusion, Cross-Site Request Forgery, and buffer overflow. In future work, the KMP algorithm will be integrated with the PHP Language and other programming languages to build a complete Secure Web Application.

References

[1] A. Hariyani and P. Dolia, "Comprehensive Review of Advanced Techniques for Mitigating SQL Injection Vulnerabilities in Modern Applications," *International Journal of Computer Applications*, 2025.

[2] O.C. Abikoye, A. Abubakar, A.H. Dokoro, O.N. Akande, and A.A. Kayode, "A Novel Technique to Prevent SQL Injection and Cross-Site Scripting Attacks Using Knuth-Morris-Pratt String Match Algorithm," *EURASIP Journal on Information Security*, Springer, 2020.

[3] S.S. Wai, "Detecting SQL Injection Attacks in Online Learning System Using Rabin-Karp Pattern-Matching Algorithm," *Master's Thesis*, University of Computer Studies (UCSY), Yangon, Myanmar, 2022.

[4] S. Pasini, G. Maragliano, J. Kim, and P. Tonella, "XSS Adversarial Attacks Based on Deep Reinforcement Learning: A Replication and Extension Study," *Journal of Cybersecurity and AI*, 2025.

[5] T. Aung, "Web Application Vulnerability Testing System Using XSS_SQL_Scanning Algorithm," *Master's Thesis*, University of Computer Studies (UCSY), Yangon, Myanmar, 2022.

[6] Acunetix, "Acunetix Web Application Vulnerability Report 2020," *Acunetix*, https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2020, Accessed: April 2025.

[7] A. Krishna, "Automated Penetration Testing," *Astra Security Blog*, https://www.getastra.com/blog/security-audit/automated-penetration-testing/, Accessed: May 2025.

[8] CWE-20: Improper Input Validation, *MITRE Corporation*, https://cwe.mitre.org/data/definitions/20.html, Accessed: March 2025.

[9] Check Point, "Top 7 PHP Security Issues and Vulnerabilities," *SpectralOps Security Blog*, https://spectralops.io/blog/top-7-php-security-issues-and-vulnerabilities/, Accessed: March 2025.

[10] OWASP Foundation, "OWASP Top 10 Web Application Security Risks – 2021," *OWASP*, https://owasp.org/www-project-top-ten/, Accessed: January 2025.

[11] OWASP Foundation, "Input Validation Cheat Sheet," *OWASP Cheat Sheet Series*, https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html, Accessed: May 2025.

[12] OWASP Foundation, "ZAP Fuzzer Add-ons," *OWASP Documentation*, https://www.zaproxy.org/docs/desktop/addons/fuzzer/, Accessed: May 2025.

Comparative Analysis of Feature-Based Phishing URL Classification Using Support Vector Machine

Nandar Lin, Phyo Phyo Wai

University of Computer Studies(Magway), Myanmar

nandarlin229627@gmail.com, phyophyowaimgy@gmail.com

Abstract

The internet has become essential to daily social and financial activities. However, it exposes users to various web-based threats, such as phishing, which can cause financial damages and loss of personal data. Phishing is a form of cyberattack where attackers disguise phishing URLs as legitimate ones to steal sensitive information. According to the Anti-Phishing Working Group (APWG), phishing attacks reached a record high in 2023, with over 4.9 million incidents reported. This study explores the use of machine learning, specifically Support Vector Machines (SVM), for classifying URLs as phishing or legitimate. This study performs a comparative analysis between two sets of features: the original 17 based features proposed by Banik et al. (2020)[1] and an extended feature set developed for enhanced phishing detection. Our results show that the extended feature set improves classification accuracy, offering a more effective approach to phishing URL detection.

Keywords*:* Phishing, Web-Based Threats, Cyberattack, Phishing Detection, Legitimate, Support Vector Machine, Lexical Features, Extended Feature Set.

1. Introduction

The internet is a vital role for communication, commerce, and education, but increasing online activity has led to a rise in cyber threats, especially phishing. Phishing involves tricking users into sharing sensitive information by impersonating trusted sources. Detecting phishing URLs is difficult due to attackers mimicking legitimate sites. Traditional rule-based methods struggle against evolving tactics, prompting the need for more adaptive solutions. Common approaches include blacklists, DNS filtering, user awareness training, and machine learning-based detection systems.

1. Blacklists: These are maintained by security companies like Google and Microsoft. Blacklists are updated regularly through automated crawlers and user reports, blocking access to these sites across various platforms. Popular tools like Google's Safe Browsing and Microsoft's SmartScreen filter rely on blacklists to warn users about potentially harmful websites.

2. DNS Filters: DNS filtering services, offered by companies like Cisco, Barracuda Networks, and Symantec, block access to malicious domains by redirecting users away from known phishing websites. These filters are effective in preventing users from reaching dangerous sites by blocking requests to malicious IPs or domains.

3. User Awareness Training: Educating users about phishing risks is a vital component of phishing prevention. Training programs teach individuals to recognize common phishing indicators.

4. Machine Learning Algorithms: Machine learning has emerged as a promising solution for phishing detection. Unlike static methods, machine learning algorithms can adapt to evolving threats by analyzing URL characteristics—such as domain names, length, and keyword presence—to identify patterns that suggest phishing attempts.

In this study, we investigate the use of machine learning for phishing URL detection, comparing the performance of Support Vector Machine (SVM) using two feature sets: the based 17 features from Banik et al. (2020)[1] and an extended set.

The remainder of this paper is organized as follows. Section 2 provides a review of related work in phishing URL detection. Section 3 describes the methodology employed in this study, including the feature sets and machine learning models. Section 4 presents the experimental results and discusses the key findings. Finally, Section 5 concludes the study.

2. Related Works

Phishing remains a critical issue in network and internet security, leading researchers to develop various methods to protect users from cyber-attacks. To prevent phishing, techniques such as blacklisting, whitelisting, machine learning, and deep learning have been employed to identify and block malicious URLs. This section reviews key studies that have used different approaches for detecting phishing sites.

Blacklisting and whitelisting are common methods for phishing URL detection. Blacklisting involves maintaining a database of known malicious URLs, as used by Google Safe Browsing API [11], which is regularly updated [4]. In contrast, whitelisting allows access only to trusted sites. Han et al. [5] introduced an Automated Individual White List (AIWL) using a Naïve Bayes classifier to alert users when interacting with unknown sites. Similarly, Jain and Gupta [7] developed a whitelist-based method that cross-references domain names and IPs, performing further checks on unlisted URLs using hyperlink features, and tested their approach on 1,525 URLs.

Blacklisting and whitelisting face limitations due to the short lifespan of phishing domains and the inability to handle zero-day attacks. To address this, Mohammad et al. [9] proposed an automated neural network using URL-based and external features (e.g., WHOIS, DNS, pop-ups). While this approach improves accuracy, it relies heavily on third-party services and is time-consuming to implement. Their method was tested on a small dataset of 1,400 URLs, highlighting the need for larger datasets to build more robust models.

A phishing URL classification system was proposed by Banik et al. [1], emphasizing lexical features of URLs. They extracted 17 features, which were prioritized using Chi-square feature selection.Four machine learning models—Random Forest, Decision Tree, Naive Bayes, and Support Vector Machine (SVM)—were tested, with the highest accuracy being achieved by Random Forest across three different datasets.

Another study[3] explores phishing URL detection through lexical structure analysis of URLs, utilizing classification models such as Extreme Gradient Boosting (XGBoost), SVM, and Artificial Neural Network (ANN) with datasets sourced from Phish Tank. This approach centers on static lexical features, achieving respective accuracies of 88%, 87%, and 88%, with XGBoost yielding the highest detection rate.

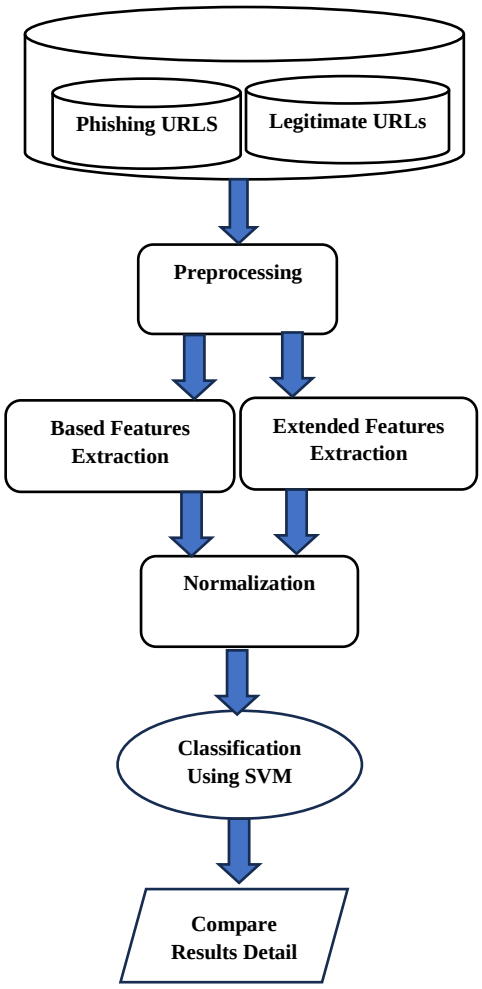


Figure1. Block Diagram of the Proposed System

3. Proposed System

We present a system for classifying phishing and legitimate URLs using lexical features and host based features of URLs. First, dataset of phishing and legitimate URLs are collected.

Lexical features and host-based of these URLs are extracted. All values of the features extracted were then normalized by applying z-score the normalization method. Experiments have been performed using machine learning technique-SVM. Then, the performances of classification method are analyzed. Figure 1 shows the block diagram of our proposed system.

3.1. Dataset Description

The legitimate URLs Dataset and Phishing URLs Dataset utilized in this study were sourced from a dataset provided by the University of New Brunswick, which is publicly available at https://www.unb.ca/cic/datasets/. The legitimate dataset consists 35,378 of records and phishing dataset consists 10,465 of records.

3.2. Preprocessing

Preprocessing is essential for preparing raw data for machine learning by improving model accuracy and reliability [8]. In this study, 10,465 phishing and 35,378 legitimate URLs were combined into a single dataset. Duplicate entries were removed to ensure data quality and representativeness. This process resulted in a clean and consistent dataset, forming a strong foundation for effective phishing URL detection.

3.3. Feature Extraction

Feature extraction is a crucial step in phishing URL detection, involving the selection of relevant characteristics from URLs. Table 1 illustrates the based feature set derived from the study by Banik et al. (2020) [1], which proposed 17 lexical features. Table 2 introduces an extended feature set designed to enhance detection accuracy. The selected features were derived and expanded based on insights from different researches [3][6][9].

In summary, feature extraction is a critical component of phishing URL detection. By combining these features, a comprehensive detection system can be built that accounts for the various techniques used by phishing attackers.

3.4. Normalization

Normalization is a crucial preprocessing technique in machine learning, particularly when algorithms are sensitive to the scale of input features, such as Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN). Without normalization, features with larger numeric ranges can disproportionately influence the learning process, resulting in biased outcomes and diminished model generalizability [10].

This study employed Z-Score Normalization to ensure all features contributed equally to the classification model and to promote faster, more stable convergence during training. As Z-Score applies only to numerical features, categorical attributes were first converted using label encoding. Then, all features were standardized using the Z-Score formula:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where: x is the original value. μ is the mean of the feature. σ is the standard deviation of the feature.

3.5. Machine Learning Techniques

Machine learning provides a framework for developing predictive models based on patterns derived from data. This study adopts the supervised learning approach, wherein labeled data is used to train a model capable of classifying URLs as phishing or legitimate. Among various supervised algorithms, the Support Vector Machine is chosen for its robustness in high-dimensional spaces and its effectiveness in binary classification[1]. SVM constructs an optimal hyperplane that maximizes the margin between classes. The decision function is defined as:

$$f(x) = \text{sign}(w \cdot x + b) \quad (2)$$

However, phishing URL data often has complex, non-linear patterns that linear models struggle to separate. To handle this, the Radial Basis Function (RBF) kernel is used to map input data into a higher-dimensional space, allowing for non-linear decision boundaries[1]. The RBF kernel is defined as:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (3)$$

where γ determines the influence of individual training samples. This kernel allows SVM to effectively capture complex relationships in the data, improving classification accuracy in phishing URL detection.

Table 1. Based Feature Set

No.	Feature Name	Description
1	Path Length Ratio	Ratio of the path segment length to the total URL length. Phishing URLs often have long or irregular paths to hide their true destination.
2	URL Length	Total length of the URL, including domain, path, and parameters. Long URLs may be used to appear legitimate or obfuscate malicious content.
3	Special Character Ratio	Proportion of special characters (%, #, &, *) in the URL. These are often used to manipulate structure or bypass filters in phishing URLs.
4	Total Suspicious Characters	Counts symbols like %, #, ^, $, *, and !. High counts may indicate an attempt to confuse or deceive users.
5	Slash Count	Number of slashes (/) in the URL. Excessive slashes can indicate suspicious directory depth, common in phishing URLs.
6	Suspicious Words Count	Number of phishing-related words (e.g., login, secure, confirm). Common phishing tactic includes such words to appear trustworthy.
7	Dot Count	Counts the number of dots (.) in the URL. Many dots may mean multiple subdomains mimicking a real domain.
8	Query Presence	Boolean feature indicating presence of a query. Often used in phishing to collect input or redirect.
9	Question Mark Count	Number of question marks (?) in the URL. Multiple question marks are unusual and can disguise the true URL intent.
10	Hyphen Count	Counts hyphens (-) in the domain. Frequently used to mimic legitimate sites (e.g., pay-pal.com).
11	@ Symbol Presence	Checks for presence of @ in the URL. Anything before @ is ignored by the browser, which is often exploited in phishing.
12	Last Character Symbol Check	Checks if the last character is a symbol other than /. Used to make URLs look legitimate or hide structure.
13	HTTP in Domain or Path	Checks if "http"/"https" appears in domain or path. Creates false appearance of security.
14	Redirection Check	Detects presence of // beyond the protocol (not at the beginning). May be used to redirect to phishing domains.
15	IP Address Presence	Boolean feature. If the domain is an IP address, it's suspicious. Legitimate sites rarely use raw IPs.
16	Unicode Character Presence	Checks for non-ASCII (Unicode) characters, used to imitate legitimate domains.
17	Port Number Presence	Detects presence of explicit port numbers. Non-standard ports may be used to evade detection.

Table 2. Extended Feature Set

No.	Feature Name	Description
1	Shortening Service	Detects if the URL uses a known shortening service (e.g., bit.ly, tinyurl.com). Phishing attackers often use shorteners to hide the final destination and trick users.
2	Top-Level Domain (TLD)	Extracts the TLD (e.g., .com,.xyz). Some TLDs are linked to phishing more often than others. Suspicious or uncommon TLDs may help identify phishing URLs.
3	Digit Count	Counts numeric digits in the URL. Excessive or strange digit use may indicate attempts to imitate legitimate sites or hide malicious intent.
4	Website Traffic	Evaluates if the website is ranked among Alexa's top 100,000. If not listed or ranked low, it may be a phishing site. Legitimate sites typically have consistent and high traffic; phishing domains are often new or unpopular.

4. Experimental Results

Performance evaluation is a critical phase in machine learning, intended to assess a model's generalization capability and its effectiveness on unseen data. In this study, a two-phase evaluation strategy was employed. Initially, the dataset was partitioned into training and testing sets using an 80:20 split, with the 20% test set representing seen data. To further assess the model's applicability in real-world scenarios, an external evaluation was conducted using independently collected datasets. This included 300 legitimate URLs sourced from a publicly available repository (https://github.com/gangeshbaskerr/) and phishing URLs obtained from the OpenPhish repository. This comprehensive evaluation framework facilitates a robust assessment of the model's reliability and generalization performance across both seen and unseen data sources.

4.1. Confusion Matrix

To provide a more detailed analysis of model performance, confusion matrices for the testing phase using both feature sets are presented in Tables 3 and 4. These illustrate true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), offering insight into classification accuracy. Results on unseen data show that the extended 21-feature set improves detection by reducing both FP and FN, enabling more accurate identification of phishing URLs and enhancing overall system reliability.

Table 3. Confusion Matrix for Based Feature Set

	Predicted (Legitimate)	**Predicted (Phishing)**
Actual (Legitimate)	287(TP)	13(FN)
Actual (Phishing)	12(FP)	288(TN)

Table 4. Confusion Matrix for Extended Feature Set

	Predicted (Legitimate)	**Predicted (Phishing)**
Actual (Legitimate)	290(TP)	10(FN)
Actual (Phishing)	5(FP)	295(TN)

4.2. Performance Metrics

Table 5 shows that the proposed classification model performs better with the extended feature set than with the based feature set. The extended features lead to higher accuracy, precision, recall, and F1-score on both seen and unseen data , indicating that the additional features contributed to improved detection of phishing URLs.

Table 5. Performance Comparison of Different Feature Sets Using SVM

Feature Set	Phase	Accuracy	Precision	Recall	F1-Score
Based Feature Set	Testing (Seen)	98.61%	98.75%	97.30%	98.00%
Extended Feature Set		99.78%	99.64%	99.74%	99.69%
Based Feature Set	Testing (Unseen)	95.83%	95.83%	95.83%	95.83%
Extended Feature Set		97.50%	97.51%	97.50%	97.50%

5. Conclusion

This study highlights the effectiveness of Support Vector Machine (SVM) in phishing URL classification, demonstrating the impact of an extended feature set on detection performance. The results indicate a substantial improvement in accuracy, underscoring the importance of comprehensive feature engineering in machine learning-based phishing detection. The incorporation of additional lexical and hosted based features enhances the model's ability to identify sophisticated phishing attempts, contributing to its reliability for real-world applications. These findings reinforce the critical role of feature extraction in optimizing classification performance and advancing phishing detection methodologies.

References

[1] B. Banik and A. Sarma, "Lexical feature-based feature selection and phishing URL classification using machine learning techniques," *Machine Learning, Image Processing, Network Security and Data Sciences*, pp. 93–105, 2020.

[2] B. Banik and A. Sarma, "Phishing URL detection system based on URL features using SVM," *International Journal of Electronics and Applied Research*, vol. 5, pp. 40–55, 2018.

[3] Cing Gel Vung and Yu Yu Win.(2023) "URL Classification Based on Lexical Features by Machine Learning (2023)". In 2023 IEEE Conference on Computer Applications (ICCA). IEEE.

[4] Gupta, B.B., Arachchilage, N.A.G., Psannis, K.E.: Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommunication Systems 67 (2), 247–267 (2017).

[5] Han, W., Cao, Y., Bertino, E., Yong, J.: Using automated individual white-list to protect web digital identities. Expert Syst. Appl. 39, 11861–11869 (2012).

[6] Hutchinson, S., Zhang, Z., Liu, Q.: Detecting phishing websites with random forest. In: Meng, L., Zhang, Y. (eds.) MLICOM 2018. LNICSSITE, vol. 251, pp. 470–479. Springer, Cham (2018).

[7] Jain, A.K., Gupta, B.B.: A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP J. Inf. Secur. 2016(1), 1–11 (2016).

[8] Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). *Data preprocessing for supervised leaning*. International Journal of Computer Science, 1(2), 111–117.

[9] Mohammad, R.M., Thabtah, F., McCluskey, L.: Predicting phishing websites based on selfstructuring neural network. Neural Comput. Appl. 25(2), 443–458 (2013).

[10]Niño-Adan, I. Landa-Torres, and E. Portillo, "Influence of statistical feature normalisation methods on K-Nearest Neighbours and K-Means in the context of Industry 4.0," *Engineering Applications of Artificial Intelligence*, vol. 111, May 2022

[11] Overview Safe Browsing APIs (v4) Google Developers. https://developers.google.com/safe browsing/v4. Accessed 18 Dec 2019

[12] Sahoo, D., Liu, C., Hoi, S.C.H.: Malicious URL detection using Machine Learning: a survey. (2017)

Transformer based Summarization on News

Aye Thinzar Ko
Faculty of Computer Science
University of Computer Studies, Taungoo
Myanmar
ayethinzarko96@gmail.com

Dr. Lwin Nyein Thu
Faculty of Computer Science
University of Computer Studies, Taungoo
Myanmar
lwinnyeinthu@gmail.com

Abstract

The length of a news article may influence reader's engagement, as lengthy texts may discourage quick consumption. To address this issue, Text Summarization has been explored to deliver concise and informative versions of longer texts. Among the two main approaches: extractive and abstractive summarization, this system focuses on extractive summarization using transformer-based pre-trained model, BART-large-cnn. The main contribution lies in the fine-tuning the existing BART-large-cnn model on a new dataset, the BBC News Summary Dataset, and comparing its performance to the original pretrained version. The quality of the generated summaries was evaluated using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric, demonstrating the effectiveness of domain-specific fine-tuning in improving summary quality and fluency.

1. Introduction

The digital era has led an overwhelming amount of news across diverse platforms, covering topics ranging from politics and economics to entertainment and sports. The sheer volume of articles often exceeds the time readers have to process them, making it challenging to stay informed.

The motivation for this study is rooted in the need to address the challenges posed by information overload. Transformer-based models offer the potential to generate high-quality, coherent summaries efficiently News summarization provides a practical solution by generating concise versions of lengthy articles, enabling readers to grasp essential information quickly and efficiently.

This study focuses on developing a Transformer-based summarization system tailored for news articles. Fine-tuning the BART-large-cnn model on BBC News Summary Dataset aims to produce fluent and concise summaries that retain the critical information from lengthy news articles. The performance of the model is evaluated using ROUGE metrics to ensure the quality and relevance of the generated summaries. The primary objectives of this study are to mitigate the challenge of information overload, enable rapid comprehension of key points, enhance the readability of news content, and reduce the time required for content consumption. Additionally, the system aims to leverage the efficiency of Transformer-based models and to compare the fine-tuned model's performance against the original pre-trained version, thereby analyzing the improvement achieved through fine-tuning.

Recent studies have demonstrated the effectiveness of various summarization tasks. A Malaysian study applied machine learning methods like SVM, Naive Bayes, Decision Tree, and Random Forest on the CNN/Daily Mail dataset, using TF-IDF and sentence-based feature vectors. Random Forest with sentence-based features achieved the best ROUGE scores, emphasizing the importance of feature selection [3]. Similarly, a study in Ireland evaluated contextual embeddings like Word2Vec, ELMo, BERT, and RoBERTa, paired with K-means clustering. BERT outperformed other embeddings, achieving the highest performance across ROUGE metrics, highlighting the advantage of transformer-based embeddings in extractive summarization tasks [4].

In the field of the abstractive summarization, researchers in India assessed models like Pipeline, BART, T5, and PEGASUS on the BBC news dataset. BART generated fluent and accurate summaries with relevant input evidence, while T5 achieved a higher ROUGE score with coherent, accurate outputs. PEGASUS produced summaries that matched closely in content but sometimes lacked completeness [5]. Another comparative study across CNN/Daily Mail, MultiNews, and XSum datasets found BART to be the most effective, achieving the highest ROUGE-1 and ROUGE-2 scores, indicating its strength in capturing essential information from texts [7].

Building upon insights from prior works, the system demonstrates improved performance in summarizing news articles by generating concise outputs that retain essential information, achieving an average content reduction of approximately fifty-five percent. Unlike traditional extractive methods, this system utilizes a fine-tuned BART-large-cnn model to produce fluent and semantically rich summaries, typically spanning eight to thirteen sentences, depending on the article's length and complexity. Comparative analysis with existing approaches reveals that the proposed model achieves higher ROUGE scores, reflecting better fluency and informativeness. These findings highlight the system's ability to preserve key content while significantly condensing article length, validating the effectiveness of domain-specific fine-tuning on a dataset like the BBC News Summary.

The remainder of this system is organized as follows: **Section 2** describes the methodology of the proposed system, including details on the BBC News Summary Dataset, preprocessing steps, the BART-large-cnn transformer model, and the evaluation metrics used. **Section 3** provides an overview of the proposed system architecture and workflow. **Section 4** presents the experimental results, including training, evaluation, and testing outcomes on the BBC News Summary dataset. Finally, **Section 5** concludes the paper and suggests potential directions for future research.

2. Methodology of Proposed System

This section outlines the methodology employed in developing an automated news summarization system using a transformer-based model. The proposed system is built using the BBC News Summary Dataset, and involves stages such as dataset preparation, preprocessing, model adaptation through fine-tuning, and performance evaluation.

2.1. BBC News Summary Dataset

The *BBC News Summary Dataset*, accessed from the Hugging Face Library, is used in this study. It consists of 2,224 news articles categorized into five categories: *Business, Sport, Politics, Tech,* and *Entertainment* [16]. It is suitable for summarization tasks due to its paired format of news articles and corresponding summaries. Each entry in the dataset consists of *File_path*, *Articles* and *Summaries* [16].

2.2. Preprocessing

In the initial phase of preprocessing, an **exploratory data analysis (EDA)** was conducted to understand the dataset's characteristics, specifically focusing on word counts within articles and summaries. The EDA focused on *Word Count Analysis, Statistical Summary*, *Data Visualization, Outlier Detection and Null Values Analysis.* After removing outliers, the dataset was reduced from 2224 to 2208 records.

The next step is **Text preprocessing** which is a crucial step to ensure that the raw input data is clean, consistent. The following steps were applied:

Text Cleaning: This step involves removing unwanted characters, special symbols, HTML tags, and extraneous whitespace from the raw text data.

Tokenization: Tokenization splits the cleaned text into smaller units called tokens, typically words or subwords. The BART tokenizer was used to convert these tokens into numerical IDs that can be interpreted by the model.

Padding: Shorter token sequences are padded with special [PAD] tokens to ensure that all inputs have a uniform length, as required by transformer-based model.

Truncation: *L*onger sequences exceeding the model's maximum token limit were truncated to ensure compatibility with model constraints and to prevent memory issues during training.

After preprocessing and removing outliers, the dataset was randomly split into two parts: **70% for training (1,545 samples)** and **30% for testing (663 samples)**. The split was performed using a **fixed random seed** to ensure reproducibility.

2.3. BART-large-cnn Transformer Model

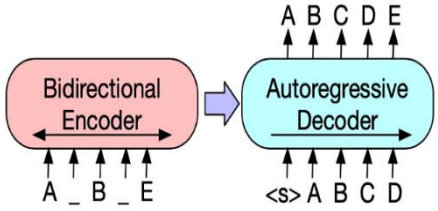


Figure 1. BART-large-cnn Transformer

The BART-large-cnn model is a transformer-based sequence-to-sequence

architecture developed by Facebook AI. It combines a bidirectional encoder (like BERT) and an autoregressive decoder (like GPT), making it highly effective for summarization tasks [13]. The overall architecture of the BART-large-cnn model is illustrated in Figure 1.

The **encoder** processes the input article in parallel, using multi-head self-attention and feed-forward layers across 12 transformer blocks. This allows it to capture comprehensive contextual information from the entire input sequence [14]. The **decoder**, also composed of 12 layers, operates autoregressively, generating one token at a time while attending to both previously generated tokens and the encoder outputs. This results in fluent and contextually aware summaries [14]. The core operation within both the encoder and decoder is the **scaled dot-product attention**, which determines the relevance of different tokens:

$$\text{Attention} = \text{Softmax}\left(\frac{Q.K^T}{\sqrt{dk}}\right) V \quad (1)$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. This mechanism helps the model weigh the importance of tokens dynamically. The model uses **multi-head attention**, enabling it to capture different types of relationships across various subspaces in parallel. Each layer also includes a **position-wise feed-forward network**:

$$FFN = \max(0, xW_1 + b_1)W_2 + b_2) \quad (2)$$

where W_1 , W_2 are learned weight matrices and b_1, b_2 are biases. Since transformers lack a built-in sense of word order, positional encodings are added to the input embeddings to retain word order information:

$$PE_{(pos,2i)} = \sin(pos/\ 10000^{2i/d}) \quad (3)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d}) \quad (4)$$

where pos: the token position and d: the dimension index.

The BART-large-cnn model consists of ***12 encoder layers and 12 decoder layers***, with each layer incorporating ***16 attention heads*** *and a* ***hidden size of 1024***. The model contains *approximately* ***406 million parameters*** [14].

2.4. Model Fine-Tuning

In the context of modern NLP, pretrained transformer models like BART are first trained on large-scale corpora (e.g., CNN/Daily Mail) using general objectives such as denoising or text reconstruction. This initial phase is known as *pretraining*. *Fine-tuning*, on the other hand, refers to taking the pretrained model and continuing training on a smaller, task-specific dataset like the BBC News Summary dataset. Fine-tuning helps the model adapt its knowledge to perform better on the specific summarization task targeted by this study.

The BART-large-cnn model was fine-tuned using the Hugging Face Trainer API with the following settings*: 6 training epochs*, *batch size of 4*, *evaluation every 500 steps*, *learning rate of 0.0001*, *weight decay of 0.01*, and *2 data loader workers*. These hyperparameters were chosen to balance performance and efficiency.

2.5. Evaluation Metrics

To assess the quality of generated summaries after modeling, this study uses **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) metrics, which measure content similarity between generated and reference summaries based on word overlap [15]. The main types of ROUGE scores are as follows:

- ***ROUGE-1:*** Measures unigram (single word) overlap [15].
- ***ROUGE-2:*** Measures bigram (two-word sequence) overlap [15].
- ***ROUGE-L:*** Measures the longest common subsequence (LCS), reflecting the sequence and main idea retention [15].

3. Overview of the Proposed System

The proposed system for news summarization follows a structured and systematic workflow consisting of four main phases: **Exploratory Data Analysis (EDA), Text Preprocessing, Model Fine-Tuning**, and **Summary Generation**. The goal of the system is to generate concise and coherent summaries of input news articles using a **pretrained transformer model (BART-large-cnn)**. The proposed system workflow is illustrated in Figure 2. The overall workflow is as follows:

Input: A news article is fed into the system.
EDA: The input data undergoes Exploratory Data Analysis to clean and analyze the dataset.
Text Preprocessing: The text is cleaned, tokenized, padded, and truncated to prepare it for the model.
Model Fine-Tuning: This involves taking a pretrained BART-large-cnn model and adapting it to a specific BBC News Summary dataset by continuing training on labeled data.
Summary Generation: The preprocessed input is passed to the fine-tuned BART-large-cnn model, which generates the final summary.
Output: The system returns a concise and coherent summary of the input article.

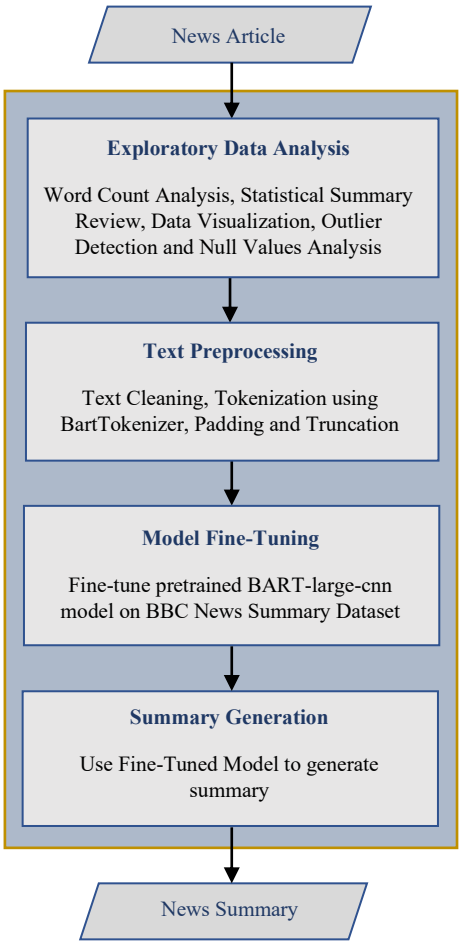


Figure 2. Overview of Text Summarization System

4. Experimental Results

This section presents the experimental outcomes and key findings from the evaluation of BART-large-cnn Transformer model fine-tuned on the BBC News Summary Dataset.

4.1. Different Training Configurations

After removing outliers, the BBC News Summary dataset contained 2208 samples, which were split into two configurations for training and testing sets: 80% train (1,779 samples) and 20% test (445 samples), as well as 70% train (1,545 samples) and 30% test (663 samples). The training results for both are presented in the following Table 1 and Table 2. Experiments were conducted on Jupyter Notebook using an NVIDIA GeForce RTX 2060 SUPER GPU.

During evaluation, the fine-tuned BART-large-cnn model generated summaries at two average lengths. However, both configurations resulted in incomplete outputs:

- ***Generated Length (123.3775):*** The model produced a summary that was truncated, ending mid-sentence without providing a complete conclusion.
- ***Generated Length (183.6214):*** Similarly, when testing with a generated length of 183.6214, the summary was also incomplete, ending abruptly.

These results suggest that additional tuning may be necessary to achieve complete and coherent summaries, as the current configurations lead to incomplete sentence generation.

Table 1. Experiments on BBC News Summary Dataset with training 80% (1779) and testing 20% (445)

Metric	Epoch 5	Epoch 3
Training Time	3:50:30	2:28:36
Training Loss	0.170095	0.233067
Validation Loss	0.156928	0.163042
ROUGE-1	0.966000	0.962800
ROUGE-2	0.938500	0.931700
ROUGE-L	0.962900	0.959700
Generated Length	123.3775	123.3775

Table 2. Experiments on BBC News Summary Dataset with training 70% (1545) and testing 30% (663)

Metric	Epoch 5
Training Time	4:09:19
Training Loss	0.033455
Validation Loss	0.103888
ROUGE-1	0.969700
ROUGE-2	0.942400
ROUGE-L	0.967400
Generated Length	183.6214

Table 3. Experiments using BBC News Summary Dataset on Google Colab for training 70% (1545) and testing 30% (663)

Epoch	Training Time	Training Loss	Validation Loss	ROUGE-1	ROUGE-2	ROUGE-L	GenLen
10	3:09:56	0. 084863	0. 191627	0. 866400	0. 843400	0. 686270	512
10	3:23:06	0.084604	0. 191600	0. 929400	0. 906300	0. 926700	512
10	3:21:57	0.084103	0. 920600	0. 920600	0. 896600	0. 917100	512
6	1:58:45	0.109449	0. 136125	0. 903600	0. 880300	0. 900900	512
6	**2:04:42**	**0.090462**	**0. 136988**	**0. 945600**	**0. 921800**	**0. 942900**	**512**

To further investigate performance, the model was trained again on 70% for training (1,545 samples) and 30% for testing (663 samples) using Google Colab with a Tesla T4 GPU. The results from this setup are summarized in Table 3.

Based on the ROUGE scores across configurations, the model trained under the final configuration highlighted in **Table 3** was selected as the **Fine-Tuned Model for Text Summarization on News**. This configuration demonstrated the highest evaluation scores, producing the most **accurate, relevant**, and **fluent summaries,** and is therefore considered the optimal model for the target summarization task.

4.2. Evaluation of Fine-Tuned Results

The following figure Figure 3. illustrates the evolution of training loss, validation loss, and ROUGE scores at various training steps during the model's training process. The x-axis represents the number of training steps, while the y-axis displays the corresponding scores and losses.

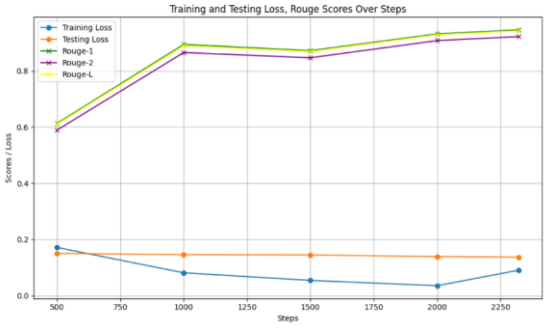


Figure 3. Illustration for Evaluation Results

The following final evaluation results shown in Table 4 demonstrate the model's effectiveness in generating fluent and accurate summaries.

Table 4. Final Evaluation Result

Metric	Value
Evaluation Loss	0.13698838651180267
ROUGE-1 Score	0.9456
ROUGE-2 Score	0.9218
ROUGE-L Score	0.9429
Generated Length	512.0
Evaluation Runtime(seconds)	173.1003
Samples Per Second	3.83
Steps Per Second	0.959
Epoch	6

4.3. Testing Results

The performance of the proposed system was evaluated using sample news articles retrieved from the testing dataset. The testing phase focused on assessing the model's ability to generate coherent, concise, and accurate summaries for unseen data. To quantify the quality of the generated summaries, ROUGE scores were employed as the primary evaluation metric. ROUGE measures the overlap between the generated summary and the reference (human-written) summary, capturing how well the system preserves important content. The ROUGE evaluation results are summarized in Table 5 and Table 6. Additionally, Figure 4 and Figure 5 provide bar chart comparisons between the original pre-trained model and the fine-tuned version, highlighting the improvements achieved through fine-tuning. The system produced the following results:

Input Article:

"Oscars race enters final furlong. The race for the Oscars entered its final stages as the deadline for voters to choose their winners passed...The 5,808 Academy voters had until Tuesday afternoon to return their ballots - any late submissions will not be included in the count. The next five days will be spent counting the voting forms and preparing the winners' envelopes. Best actor nominee Leonardo DiCaprio is to present a statuette for the first time at the LA ceremony on Sunday...The 30-year-old actor, who is nominated for playing Howard Hughes in The Aviator, will join other hopefuls such as co-star Cate Blanchett, Natalie Portman and Kate Winslet as Oscar presenters. The only people who will know the Oscar winners before they are revealed at the ceremony will be the auditors who are in charge of looking after the ballot count...After collating the results,

they are responsible for sealing the results in the famous golden envelopes which will be revealed by a host of celebrity presenters at the ceremony. Former Academy Award winners Gwyneth Paltrow, Dustin Hoffman and Halle Berry will also present prizes. The event at the Kodak Theatre will be attended by 3,300 people, including some of the best-known names in film, and organisers say they expect it will be watched on television by one billion people around the world. One current concern is the torrential rain which has lashed Los Angeles for the past week, flooding suburbs and causing mudslides. It is hoped the forecast for Sunday, for cool weather but no rain, will prove accurate. ""The last time it rained on Oscars night was in the mid-to-late 1980s,"" said Oscars communications director John Pavlik. ""We have had rain up until the day before the show many times, but for some reason the Oscar gods always shine on Sunday and we hope they will do so again this year,"" he added."

Reference Summary:

The only people who will know the Oscar winners before they are revealed at the ceremony will be the auditors who are in charge of looking after the ballot count."We have had rain up until the day before the show many times, but for some reason the Oscar gods always shine on Sunday and we hope they will do so again this year," he added.Best actor nominee Leonardo DiCaprio is to present a statuette for the first time at the LA ceremony on Sunday.The race for the Oscars entered its final stages as the deadline for voters to choose their winners passed.It is hoped the forecast for Sunday, for cool weather but no rain, will prove accurate.The 5,808 Academy voters had until Tuesday afternoon to return their ballots - any late submissions will not be included in the count.

Summary generated by Original Bart-large-cnn Transformer Model

5,808 Academy voters had until Tuesday afternoon to return their ballots. Next five days will be spent counting the voting forms and preparing the winners' envelopes. Best actor nominee Leonardo DiCaprio is to present a statuette for the first time at the LA ceremony on Sunday.

Summary generated by Fine-tuned Model

The only people who will know the Oscar winners before they are revealed at the ceremony will be the auditors who are in charge of looking after the ballot count."We have had rain up until the day before the show many times, but for some reason the Oscar gods always shine on Sunday and we hope they will do so again this year,"" he added."The race for the Oscars entered its final stages as the deadline for voters to choose their winners passed.Best actor nominee Leonardo DiCaprio is to present a statuette for the first time at the LA ceremony on Sunday.The 5,808 Academy voters had until Tuesday afternoon to return their ballots - any late submissions will not be included in the count.

Table 5. ROUGE Scores for Original BART-large-cnn Transformer Model

Metric	Precision	Recall	F1-score
ROUGE-1	0.8500	0.2800	0.4200
ROUGE-2	0.6700	0.2200	0.3300
ROUGE-L	0.5300	0.1700	0.2600

Table 6. ROUGE Scores for Fine-Tuned BART-large-cnn Model

Metric	Precision	Recall	F1-score
ROUGE-1	1.0000	0.8900	0.9400
ROUGE-2	0.9800	0.8700	0.9300
ROUGE-L	0.8700	0.7700	0.8200

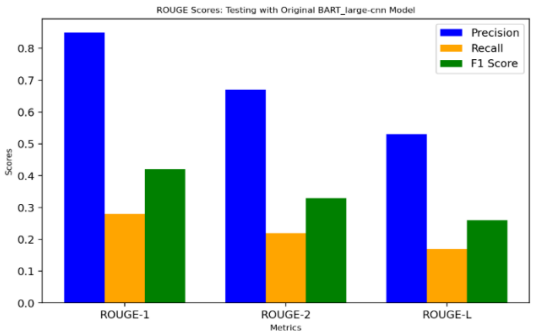


Figure 4. ROUGE Score Testing for Original Model

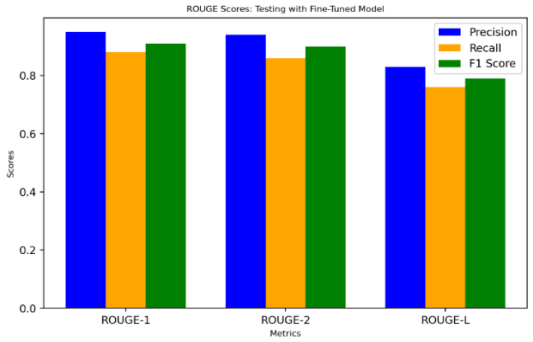


Figure 5. ROUGE Score Testing for Fine-Tuned Model

5. Conclusion

This study explored the use of transformer-based models for text summarization, specifically a particular focus on news articles. It provides and essential tool for efficiently conveying key information. By utilizing a pretrained transformer model, namely BART-large-cnn, this study provided transfer learning to enhance summarization performance through fine-tuning on BBC News Summary dataset.

This fine-tuned model demonstrated significant improvements over the original version, producing more coherent and contextually accurate summaries, while reducing the original content by approximately *fifty-five percent* on average. This enhancement is reflected in improved ROUGE scores ***(ROUGE-1: 0.945600, ROUDE-2: 0.921800, ROUGE-L: 0.942900)*** indicating a closer alignment with human-generated reference summaries.

A ROUGE-1 score of 0.9456 indicates the summaries retain about 94.56% of key words from the reference. The ROUGE-2 score of 0.9218 shows the model captures important two-word phrases, enhancing fluency. The ROUGE-L score of 0.9429 reflects strong preservation of sentence structure and readability. Collectively,

these scores confirm that the fine-tuned model produces accurate, fluent and natural summaries.

Future work could explore multi-document summarization, and alternative transformer models such as PEGASUS or T5. Additionally, multilingual training may further enhance summary quality and expand real-world applicability. Moreover, cross-validation will be considered in future experiments to ensure robustness and minimize bias in model performance due to specific data partitions.

6. References

[1] IBM, "Text Summarization," 2023.

[2] A. Health, "Extractive vs. Abstractive Summarization in Healthcare," *Medium,* March 2021.

[3] O. V. J. C. X. a. K. K. W. Hew Zi Jian, "Text Summarization for News Articles by Machine Learning Techniques," *Applied Mathematics and Computational Intelligence (AMCI),* 31 December 2022.

[4] S. V. Karkada, "Extractive Text Summarization of News Reports Leveraging Transfer Learning Contextual Embedders," 15 August 2022.

[5] D. C. A. a. R. K. Anushka Gupta, "Automated News Summarization Using Transformers," *arXiv,* 23 April 2021.

[6] S. L. S. a. D. S. S. Sathya, "Extractive - Abstractive Summarization Using Transformers: A Hybrid Approach," *Journal of Pharmaceutical Negative Results,* 2022.

[7] M. A. a. R. B. Ambrish Choudhary, "Comparative Study on News Summarization using various Transformer Based Models," *IEEE,* 17 April 2023.

[8] IBM, "Recurrent Neural Networks," 2024.

[9] D. Kalita, "A Brief Overview of Recurrent Neural Networks (RNN)," *Analytics Vidhya,* p. March, 2022.

[10] shipra_saxena, "Introduction to Long Short-Term Memory (LSTM)," *Analytics Vidhya,* p. March, 2021.

[11] P. Srivatsavaya, "LSTM Implementation, Advantages, and Disadvantages".

[12] IBM, "Transformer Model".

[13] Dataloop, "BART-Large-CNN Model".

[14] ParamRaval, "Transformers BART Model Explained," *ProjectPro.*

[15] E. Kızılırmak, "Text Summarization: How To Calculate Rouge Score," *Medium.*

[16] H. Face, "BBC News Summary Dataset".

University Students' Stress Level Classification Using Random Forest Model

Thon Thant Thant Thaung
University of Computer Studies (Taungoo)
Taungoo, Myanmar
thonthantthantthaung16@gmail.com

Dr. Hnin Pwint Phyu
Faculty of Computer Science
University of Computer Studies (Taungoo)
Taungoo, Myanmar
hninpwintphyu.ucs@gmail.com

Abstract

The current education system and the intense competition lead to increased anxiety and stress among university students. Primary sources are academic pressures such as coursework, exams, and projects, along with external pressures like peer influence and family expectations. A dataset of 450 responses was collected from students at the University of Computer Studies (Taungoo) using Google Form survey, based on the Student Stress Inventory (SSI), including 40 stress-related questions. In this proposed system, stress levels are categorized into mild, moderate, or severe, and stress factors are identified into Physical, Interpersonal Relationship, Academic and Environmental categories by using Random Forest classification model. This model achieved an accuracy of 88.14%, which improved to 96.94% with oversampling techniques. Furthermore, descriptive statistics indicate that approximately 60% of students suffer moderate stress, with academic stress being the most impactful factor, affecting around 64% of the students.

1. Introduction

University students suffer stress from exams, deadlines, new environments, and concerns about the future. Stress is a natural response that can encourage students to perform under pressure. Small amount of stress can be good, because it forces students to work hard and do their best during exams [15]. However, excessive stress leads to mental health issues and affects academic performance. Traditionally, stress assessment involves consulting psychiatrists. But, this is often costly and many students can't afford too. To address this, this proposed system aims to provide a more data-driven approach to understanding and addressing student stress, contributing to improved mental health support in academic institutions.

Machine learning is the branch of Artificial Intelligence based on developing models and algorithms that let computers learn from data and improve through previous experience without being explicitly programmed [16]. Among various algorithms, Random Forest model has been selected due to its high accuracy, resistance to overfitting, and interpretability through feature importance scores. Its strong performance in stress classification tasks has been demonstrated in multiple related studies [1, 2, 3, 4] , where this model often outperformed or matched other models such as Logistic Regression, Decision Tree and SVM.

2. Related Work

Binny et al. [1] conducted research on stress classification among college students one week before and after semester exams using the Perceived Stress Scale (PSS). Linear Regression, Naïve Bayes, Random Forest, and Support Vector Machine (SVM) algorithms were used to classify students as Highly Stressed, Stressed, or Normal based on 150 students' responses. Among them, SVM achieved the best accuracy of 85.71% using 10-fold cross-validation. However, the small dataset leads to the reduction of generalizability of analysis results.

Mehedi Firoz et al. [2] researched the percentage of students with emotional pressure. Data were collected from 2116 students across Bangladesh universities and Decision Tree, Random Forest, SVC, KNN, and Multinomial Naïve Bayes classifiers were applied to classify students as "Depressed" or "Non-Depressed".

Due to class imbalance, an oversampling technique was used. Although Random Forest and Decision Tree achieved the highest accuracy of 99%, this approach may limit to the models' performance due to the lack of real-time testing data and binary classification may lead to an oversimplification of stress levels.

Raghav V. Anand et al. [3] developed an ensemble learning model for classifying student stress into highly stressed, manageable stress, and no stress. Decision tree, Random Forest classifier, AdaBoost, Gradient Boost, and ensemble learning algorithms with various combinations were implemented. A dataset of 197 teenage students was used and oversampling was used to balance the classes. Random Forest model was the best and it gave 86.96% of accuracy. The ensemble of RF, DT and AB models reached 93.45% accuracy after applying five-fold cross-validation. But the evaluation of students' stress may be complex for capturing due to the small number of survey questions.

This paper is one of the key inspirations. Salma S Shahapur et al. [4] developed a predictive model to categorize student's stress levels and support early interventions based on self-reported data, academic performance, and study load. The dataset was from Kaggle containing more than 6000 samples. Multiple machine learning techniques, including Logistic Regression, Random Forest, K-Nearest Neighbors, Gradient Boosting, and Decision Tree were used to classify "Low", "Medium" and "High" categories. Random Forest model obtained the best accuracy at 95%. A key difference is that the proposed system not only classifies the stress levels but also identifies the stress factors affecting students using a real-time dataset.

Khaing Nyein Thant et al. [5] made research to measure students' stress and internet addiction levels using the Internet Addiction Test (IAT) and the Student Stress Inventory (SSI) based on 2612 students' responses. Their findings revealed that 28% of students experienced internet addiction and stress level was mild. Even though this paper focuses more on psychological aspects, it still offers valuable insights for our research. Due to difficulties in consulting psychiatrists for expert validation, SSI (2020 Edition) was applied as a standardized tool for stress assessment, ensuring consistency with prior studies.

While previous studies mainly concentrated on classifying stress levels, stress factors identification was extended to capture the causes of students' stress in this proposed system. To improve the dataset quality, oversampling and undersampling techniques were applied and Gridsearch CV was also used to optimize the model's parameters. To offer deeper insights, the model was tested on real-time dataset and five-fold cross validation was utilized to enhance model's performance.

3. Overall Architecture Design of the Proposed System

Figure 1 illustrates the overall architecture design of the proposed system. This proposed system utilizes student stress dataset as input and applies Random Forest classification model to classify stress levels into mild, moderate or severe stress. Moreover, the most impactful stress factor is also identified into physical, interpersonal relationship, academic and environmental stress.

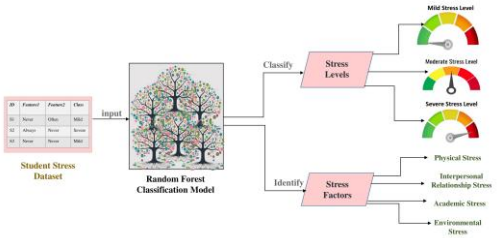


Figure 1. Overall architecture design of the proposed system

3.1. Overall Design of the Proposed System

Figure 2 demonstrates the overall design of the proposed system. The process starts by loading students' stress responses dataset and then, data are preprocessed. Next, stress levels are classified by applying Random Forest classification model and feature importance scores are used to calculate for identifying stress factors. These scores are then compared to identify the most impactful factor. Finally, the classified stress level and stress factor are displayed for each student.

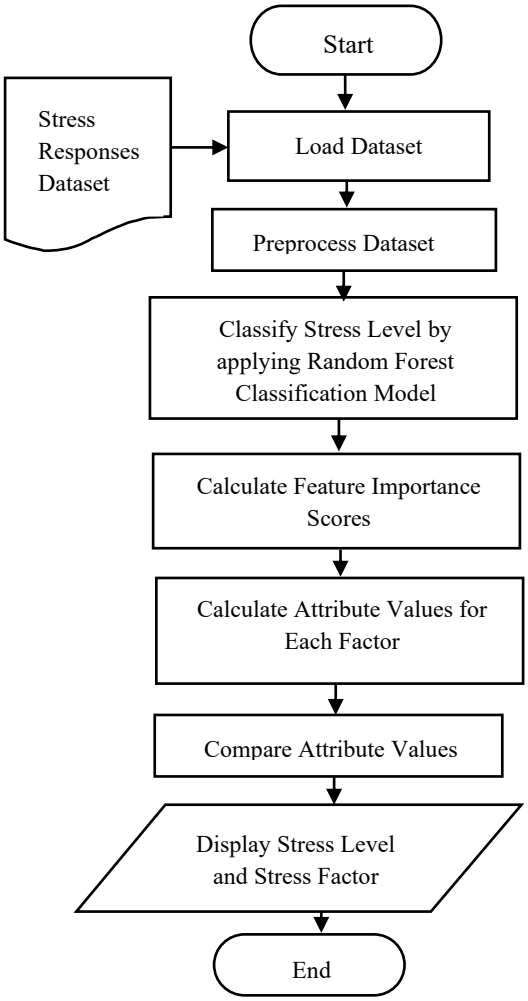


Figure 2. Overall workflow of the proposed system

4. Methodology

In this proposed system, a self-reported survey was created for data collection and the SSI was applied for stress assessment. Min-max normalization was also performed to scale the data to a specific range between 0 and 1, to prevent large numerical differences from influencing the results [6]. Min-max scaling is shown in Equation (1). *X'* means normalized value, *x* means original value, *max(x)* means maximum value of *x*, min(x) means minimum value of x, the value of new_max is 1 and the value of new_min is 0.

$$X' = \frac{x - \min(x)}{\max(x) - \min(x)} (new\_max - new\_min) + new\_min \quad (1)$$

To address class imbalance, both oversampling and undersampling techniques were applied. And Random Forest classifier was employed for stress level classification and stress factor identification. It creates multiple decision trees during training and outputs the mode of the predictions by individual trees to get stable predictions [10]. Gini index is used to measure the impurity of a node when developing decision trees [9].

$$\text{Gini} = 1 - \sum_{i=1}^{j} P(i)^2 \quad (2)$$

Where *j* is total number of classes, *P (i)* is the relative frequency of class *i*. GridSearchCV was also used for hyperparameter tuning to select the optimal configuration of the model [12]. Five-fold cross-validation was also applied to evaluate the model's generalizability and ensure consistent performance [13].

The performance of classifier was measured by using performance metrics. To define these metrics, the following terms were used [7]: True positives (TP), True negatives (TN), False positives (FP) and False negatives (FN). Precision means exactness – the percentage of tuples that the classifier labeled as positive are actually positive [7].

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Recall means completeness – the percentage of positive tuples did the classifier label as positive [7].

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

Accuracy is the percentage of test set tuples that are correctly classified. The perfect score is 1.0 [7].

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

F1 Score is harmonic mean of precision and recall [7].

$$\text{F1 Score} = \frac{2*precision*recall}{precision+recall} \quad (6)$$

4.1. Data Collection and Data Preprocessing

Following the approach of Khaing Nyein Thant et al. [5], who utilized the SSI (2015 Edition) as a standardized tool, this system applied the updated SSI (2020 Edition) to ensure consistency. Data were gathered from about 450 students of University of Computer Studies (Taungoo) in Myanmar using online Google Form survey. SSI measures stress levels in higher education institutions, focusing on the academic and campus stress. It consists of 40 questions based on the General Adaptation Syndrome Theory and the Environmental Stress Theory. In

constructing the dataset, the primary stress factors were based on the four subscales defined by SSI: Physical Stress, Interpersonal Relationship Stress, Academic Stress and Environmental Stress. The responses were scored on a scale from 1 to 4 (1 = Never, 2 = Somewhat Frequent, 3 = Frequent and 4 = Always). The total scores of each student were used to categorize stress levels based on SSI scoring guideline [8]: Mild Stress: 40-80, Moderate Stress: 81-121 and Severe Stress: 122-160.

The dataset includes about 450 students from various academic levels. In the first year (second semester), 99 students were enrolled under CST (Computer Science and Technology) program. From the second year onward, the students are categorized into CS (Computer Science) and CT (Computer Technology) majors. In second year (second semester), there was 96 CS and 17 CT students, while 58 CS and 18 CT students were enrolled in third year (first semester). There are 35 CS students and 11 CT students in third year (second semester) as well as 55 CS students and 8 CT students in fourth year (second semester). The final year contained 42 CS students and 7 CT students. At the master's level, the dataset contains two Software Engineering (SE) students and one Knowledge Engineering (KE) students.

In data preprocessing, the survey was designed to answer all questions on each page for preventing incomplete responses. However, duplicate submissions from the same participant were removed to maintain data quality. Then, ordinal encoding was applied to convert categorical data into numerical data. Equation (1) was used to rescale the values between 0 and 1 to ensure consistency in data distribution. For the analysis of the minority class, Synthetic Minority Over-Sampling Technique (SMOTE) was used to solve the underrepresented class (Severe Stress) by generating synthetic samples, while Near Miss was also applied to reduce the size of the minority classes (Mild and Moderate Stress) to balance the dataset for improving classification accuracy [11].

5. Model Training and Evaluation

The dataset was split into 70% for training the model and 30% for testing performance. Random Forest classifier with custom hyperparameters was trained on the preprocessed original dataset and two balanced datasets. Performance was evaluated by using performance metrics. GridSearchCv was applied to fine-tune hyperparameters on balanced datasets and five-fold cross validation was also utilized on all datasets to ensure model stability and prevent overfitting [13].

Table 2 illustrates the comparison results of the classifier based on three datasets. Original dataset contains real-time preprocessed data, while oversampled and undersampled datasets applied balancing techniques respectively. Train-test splitting and five-fold cross-validation evaluation techniques are used to assess performance metrics. Bar charts were used to visualize the insights which shows the distribution of stress level based on gender, academic major and class year using original dataset. Since the total number of female students (283) is higher than male students (167), percentage-based comparisons provide a clearer understanding of stress distribution.

Table 2. Performance comparison results of Random Forest classifier under different evaluation methods

Dataset	Evaluation Method	Accuracy	Precision	Recall	F1 score
Original	Train-Test Split	88.148%	82.702%	83.654%	83.151%
Oversampled	Train-Test Split	96.939%	96.776%	96.085%	96.402%
Undersampled	Train-Test Split	80.488%	80.855%	83.855%	83.855%
Original	Five-Fold Cross-Validation	89.437%	87.142%	90.813%	88.263%
Oversampled	Five-Fold Cross-Validation	98.904%	98.252%	99.060%	98.613%
Undersampled	Five-Fold Cross-Validation	87.252%	88.788%	86.373%	86.016%

In Figure 3, about 63.5% of females experienced moderate stress, compared to 56.0% of males. Severe stress was more in males (13.3%) than females (9.2%).

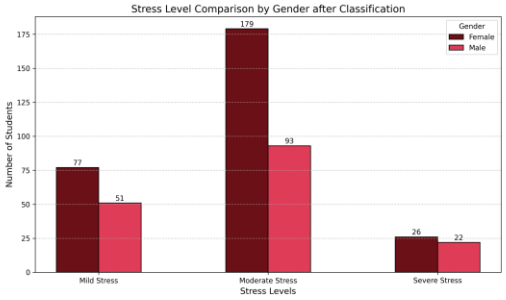


Figure 3. Stress level analysis based on gender

As shown in Figure 4, CT students had the highest proportion (76.6%) of moderate stress. CS students experienced higher mild stress (59.8%), while severe stress was the least in all majors. Sample sizes in SE and KE were too small for meaningful conclusions.

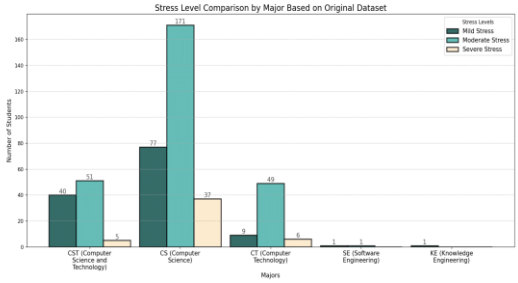


Figure 4. Stress level analysis based on major

In Figure 5, moderate Stress was the most prevalent level. In First and Second Year (Second Semester), over 60% of students experienced moderate stress level, indicating slightly higher stress levels in the earlier years. In the Third Year (Second Semester) and Final Year, no students experienced Severe Stress.

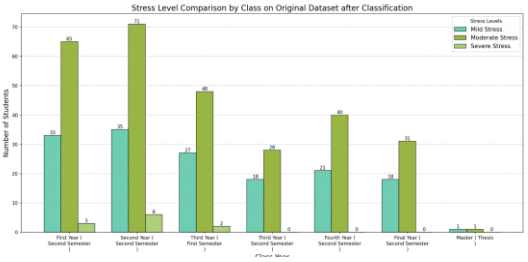
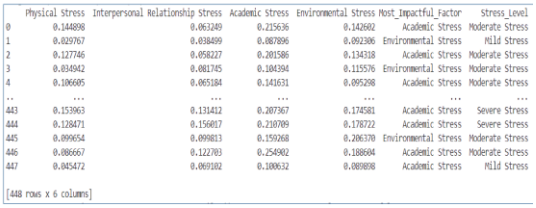


Figure 5. Stress level analysis based on academic year

5.1. Identifying Stress Factors based on Feature Importance Scores

Feature importance refers to techniques that calculate a score for all the input features and it determines how much specific variables influence the predictions of a machine learning model [14]. Feature importance scores were obtained directly after applying Random Forest model. For each student, the scaled values of their responses original dataset were multiplied by the corresponding feature importance scores. These weighted values were summed within each predefined stress factor group, compared and identified the highest value as the most impactful stress factor for that student. Total scores are shown in Figure 6.



	Physical Stress	Interpersonal Relationship Stress	Academic Stress	Environmental Stress	Most_Impactful_Factor	Stress_Level
0	0.144898	0.063249	0.215636	0.142602	Academic Stress	Moderate Stress
1	0.029767	0.038499	0.087896	0.092306	Environmental Stress	Mild Stress
2	0.127746	0.058227	0.201586	0.134318	Academic Stress	Moderate Stress
3	0.034942	0.081745	0.104394	0.115576	Environmental Stress	Moderate Stress
4	0.106605	0.065184	0.141631	0.095298	Academic Stress	Moderate Stress
..	...	...	...	...	...	...
443	0.153963	0.131412	0.207367	0.174581	Academic Stress	Severe Stress
444	0.128471	0.156017	0.210709	0.178722	Academic Stress	Severe Stress
445	0.099654	0.099813	0.159268	0.206370	Environmental Stress	Moderate Stress
446	0.086667	0.122703	0.254902	0.188604	Academic Stress	Moderate Stress
447	0.045472	0.069102	0.100632	0.089898	Academic Stress	Mild Stress

[448 rows x 6 columns]

Figure 6. Total feature importance scores of each student

Figure 7 shows that the most impactful stress factor for students is Academic Stress (43.97%) and the second is Environmental Stress (41.29%). And Physical stress is 10.27%, followed by Interpersonal Relationship Stress (4.46%). These results describe that academic and environmental pressures primarily contribute to students' stress.

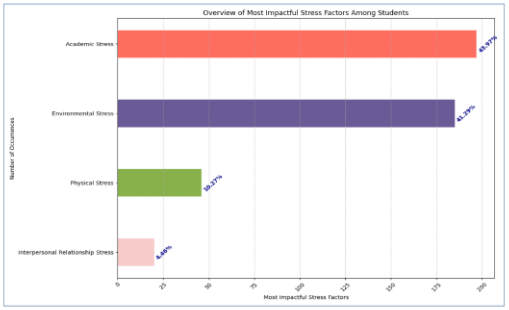


Figure 7. Analysis result of stress factor distribution

6. Conclusion and Future Work

In this study, a Random Forest classifier was employed to classify university students' stress levels into mild, moderate, and severe categories and identify stress factors into academic,

environmental, physical, and interpersonal relationship stress. This model achieved an accuracy of 88.15% improving to 96.94% with oversampling techniques. After applying five-fold cross-validation, the accuracy increased to 89.44% on the original dataset and 98.9% on the oversampled dataset. The findings indicate that students mostly experience moderate stress level and academic stress is the most impactful stress factor. Educational institutions can use these results to offer targeted support and to manage stress effectively. This analysis supports to improve the mental and physical health of students in my university by guiding students towards healthier coping mechanisms.

The current system is deployed with a single machine learning model and is based on 450 students' responses from University of Computer Studies (Taungoo). In the future, this system can be expanded to other higher education institutions and other universities, utilizing a large amount of training and testing dataset to validate its accuracy. Although the Random Forest model was chosen in this system due to its strong performance in prior research, this system didn't include a comparative analysis with other machine learning algorithms. Therefore, future improvements will involve employing additional models to help confirm the effectiveness of the Random Forest model.

References

[1] B.S, D.K. Lal, C. Mathew, and J. Nazar, "Mental Stress Detection in Students using Machine Learning Algorithms", *Journal of Emerging Technologies and innovative Research,* www.jetir.org, July 2021, n.p.

[2] Firoz, M., M.M. Islam, M. Shidujaman, A. Islam, and Md.T. Habib, *University student's mental stress detection using machine learning,* Proceeding of SPIE 12779, SPIE, Bellingham, WA, 2023, pp. 1-13.

[3] R.V. Anand, A.Q. Md, S. Urooj, S. Mohan, M.A. Alawad, and A. C, "Enhancing Diagnostic Decision-Making: Ensemble Learning Techniques for Reliable Stress Level Classification", *Diagnostics*, MDPI, Basel, Switzerland, 2023, vol. 3, pp. 1- 25.

[4] S.S Shahapur, P. Chitti, S. Patil, C.A. Nerurkar, V.S Shivannagol, V.C. Rayanaikar, V. Sawant, and V. Betageri, "Decoding Minds: Estimation of Stress Level in Students using Machine Learning", *Indian Journal of Science and Technology 17(19)*, Indian Society for Education and Environment (iSee), India, 2024, pp. 1-11.

[5] K.N. Thant, and N.N. Khaing, "A Study on Stress and Internet Addiction of Myanmar University Students", *Journal of Myanmar Academy of Arts and Science",* Myanmar Academy of Arts and Science, Myanmar, January 2020, pp. 1- 10.

[6] P.J. Muhammad Ali, "Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements", *ARO – The Scientific Journal of Koya University*, Kurdistan Region – Iraq, 2022, pp. 1-7.

[7] Han, J., M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques,* Morgan Kaufmann, Waltham, 2012.

[8] Arip, M.A.S.M., D.N. Kamaruzaman, A. Roslan, A. Ahmad, *Student Stress Inventory (SSI), Edition 2020,* Sultan Idris Education University, Malaysia, 2023.

[9] S. Tangirala, "Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm", *IJACSA,* The Science and Information Organization, New York, USA, pp. 1-9.

[10] J.K. Jaiswal, R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression", *Proceedings of World Congress on Computing and Communication Technologies (WCCT)*, IEEE, 2017, pp. 1-5.

[11] P. Canuma, "How to Deal with Imbalanced Classification and Regression Data", Neptune.ai Blog, Neptune, 2023. [Online]. Available: https://neptune.ai/blog/how-to-deal-with-imbalanced-classification-and-regression-data

[12] Great Learning Editorial Team, "Hyperparameter Tuning with GridSearchCV", Great Learning Blog, Great Learning, 2024. [Online]. Available: https://www.mygreatlearning.com/blog/gridsearchcv/

[13] V. Chuganl, "A Comprehensive Guide to K-Fold Cross Validation", Datacamp, 2024. [Online]. Available: https://www.datacamp.com/tutorial/k-fold-cross-validation

[14] T. Shin, "Understanding Feature Importance in Machine Learning", Built In, Built In, 2024. [Online]. Available: https://builtin.com/data-science/feature-importance

[15] NHS, "Student Stress", NHS, London, 2024. [Online]. Available: https://www.nhs.uk/mental-health/children-and-young-adults/help-for-teenagers-young-adults-and-students/student-stress-self-help-tips/

[16] C.R. China, "Five machine learning types to know", IBM, 2025. [Online]. Available: https://www.ibm.com/think/topics/machine-learning-types.

Multi-Class News Classification using Fine-Tune BERT Model

Thiri Zaw, Nan Sawkalayar
University of Computer Studies, Taunggyi
thirizaw@ucstgi.edu.mm, sawkalayar@ucstgi.edu.mm

Abstract

With the rapid expansion of digital news platforms, millions of articles are published each day, which increases the need for accurate and automatic classification for better organization, retrieval, and consumption. Most of the traditional machine learning approaches use TF-IDF method for feature extraction and text preprocessing. TF-IDF method has limited effectiveness because it is difficult to capture the semantic and contextual richness of text for multi-class news categorization. To address the drawback of traditional methods, this system proposed a fine-tuned BERT model which have deep semantic understanding and bidirectional context-awareness, for multi-class news classification. The system in this paper achieved the accuracy 97 %, 98%, and 99%. The data used in the proposed model is trained and tested on BBC dataset which consists of 2225 news articles with five categories (Business, Entertainment, Politics, Sport, and Tech). Moreover, additional news data 21578 including Myanmar news is collected. By adding new category class and name this class as "Travel" to this dataset as the contribution of the proposed system.

1. Introduction

Nowadays, the large amount of news articles rises. Therefore, effective classification is essential for huge amount of data. Multi-class news classification is essential for organizing and retrieving large amount of content, supporting media monitoring and more. Traditional methods only concentrate on word frequency characteristics and totally ignore the contextual information contained in text [6,7]. There has been a complete transition in recent years from these traditional text classification techniques to far stronger state-of-the-art Deep Learning (DL) based techniques [8,9]. Among various deep learning models, Bidirectional Encoder Representation (BERT) is one of the state-of-the-art solutions for natural language processing (NLP) tasks. BERT made significant improvements to NLP in 2018 by introducing the transformer architecture and using bidirectional context awareness [5].

This paper focuses on system that using a fine-tuned BERT model for the multi-class news classification. The proposed system is trained and validated on BBC dataset that includes 2225 data with five categories including business, entertainment, tech, politics, and sport. Moreover, additional news data is collected. The dataset consists of 21578 news data and putting additional news category name it as "Travel". The news data are gathered from multiple news resources including Eleven Media Group, Myanmar Now website, the Irrawaddy-Covering Burma and so on. Moreover, hybrid dataset is created by adding collected news data to BBC News Dataset is trained and tested. The implementation outcomes achieve a satisfactory level of recognition accuracy.

2. Related Work

A.Jeelani and A.Muqeem highlighted the rising difficulty of data management of large amounts of unstructured online news data. The authors described possible automatic text classification methods, and collected data from various Indian news website. They used a Naïve Bayes classifier and their study examines different classification method and eventually achieved 93% accuracy, exceeding the models of Support Vector Machines (SVM), Logistic Regression (LR), K-Nearest Neighbours (KNN) [1].

M.Jannat, E.Hossain, M.Hoque and A.Rahaman described the burden of multi-class short text categorization in Bengali language by constructing an ensemble of deep learning classifiers. They developed an ensemble model composed of a neural network (NN),

convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and finally a bidirectional gated recurrent unit (BiGRU), that uses corpus of more than 0.13 million labelled Bengali news headlines, each labelled into six classes. Their system outperformed the individual models and achieved a weighted F1-score of 84.4 [2].

Y.K. Goel and S.D. Samantaray explored a Bidirectional Long Shot-Term Memory (Bi-LSTM) model to conduct multi-label news classification. The authors collected news article headlines online in real-time and applied Bi-LSTM to classify them with an accuracy rate of 97.91%. The authors compared an assessment of the model's performance against well-known machine learning algorithms (Random Forest, Support Vector Machines, and Naïve Bayes), showing that deep learning approaches outperform classical approaches [3].

O.Blanc examines a multi-class fake news detection paradigm and base it off of classifiers derived from BERT. They emphasized the potential for transfer learning methods and deep learning text classification, and how these could address some of the issues with data imbalance and limitations around sequence length. They also stated the importance of having stronger data cleaning and processing protocols, and they demonstrated that although accuracy within the validation datasets was high, the performance decreased in the test dataset due to a change in the data distribution [4].

3. Methodology

3.1. Proposed System Design

In this paper, a well-known deep learning model called BERT (Bidirectional Encoder Representation from Transformers) is proposed to present text news classification. The proposed system architecture is trained and tested on three datasets. Firstly, the dataset is preprocessing. Then the dataset is split into training and testing dataset for model training, and classification process is done by using neural network. The proposed system architecture is illustrated in Figure 1.

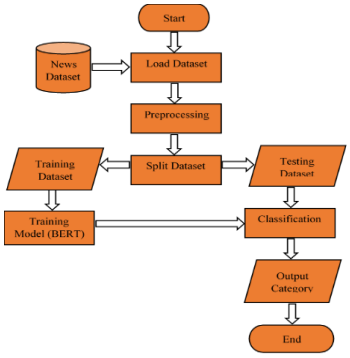


Figure 1. Proposed System Design

3.2. Datasets

3.2.1. BBC Datasets

The BBC datasets is the well-known news dataset from Kaggle website. It includes 2225 news articles with five categories. There are 510 documents for business, 386 documents for entertainment, 511documents for politics, 417 documents for sport, and 401 documents for tech. Table 1 shows the dataset of class name and number of documents in that BBC news dataset.

Table 1. BBC Dataset

Category	Number of Documents
Business	510
Entertainment	386
Politics	511
Sport	417
Tech	401

3.2.2. Own News Dataset

Own news dataset is created and then collect the data from multiple news resources including Myanmar news. This dataset includes the total number of documents 21578 and adding one newer category called "Travel". Most of the news are collected from Global News Light of Myanmar website, Eleven Media Group, Myanmar Now website, the Irrawaddy- Covering Burma and Southeast Asia website, Mizzima, Myanmar Newspapers and News Sites, Burma News International, Myanmar Travel News Website and 7 days News Myanmar. There have 5926 documents in business, 1414 documents in entertainment, 5673 documents in sport, 2003 documents in politics, 5557 documents in tech and 3000documents in travel. Table 2 shows the categories and number of documents in that class.

Table 2. Own News Dataset

Category	Number of Documents
Business	5926
Entertainment	1414
Politics	2003
Sport	5673
Tech	5557
Travel	3000

3.2.3. Hybrid Dataset

The hybrid dataset is created which include (23803 documents). This dataset could be formed by adding the BBC news dataset (2225 documents) and own news dataset (21578). This dataset includes six categories. There are 6436 documents in business, 1805 documents in entertainment, 4184 documents in sport, 2420 documents in politics, 5958 documents in tech and 3000 documents in travel. Table 3 shows the categories class in the dataset and the number of documents in that class.

Table 3. Hybrid Dataset

Category	Number of Documents
Business	6436
Entertainment	1805
Politics	2420
Sport	4184
Tech	5958
Travel	3000

3.3. Bidirectional Encoder Representation from Transformers (BERT)

Deep learning is a specialized branch of machine learning coupled with artificial intelligence. Classification tasks can be taught to deep learning models in several fields, including photographs, texts, audios, etc. To classify news categories, BERT which is a type of deep neural networks is used in this paper. BERT, which is short for "Bidirectional Encoder Representations from Transformers" was proposed in 2018. The block diagram of the proposed system in shown in figure 2.

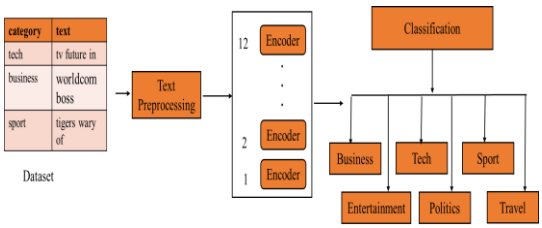


Figure 2. Block Diagram of Proposed System

There are two main works in BERT text preprocessing and pre-training, and the detail of their work are shown in the following:

3.3.1. Text Preprocessing

Preprocessing steps is very important in machine learning because it speed up the model to get the better accuracy result. The text contents are converted into lower case by using bert-base-uncase, and tokenized the sentence with bertTokenizer. Then, the additional tokens such as [CLS], [SEP], [PAD] tokens are added. The tokens are substituted with the token's ids and is converted to vector or embeddings. Input embeddings of BERT's encoder layer is a compound embedding that is linear sum of token embedding, segment embedding, and positional embedding. Token embedding transforms words into a fixed-size vector form, that is a vocabulary index for each token. Segment embeddings of size (1, n, 768) are used as vector forms to assist BERT in recognizing the distinction between paired input sequences. In addition, transformer positional embeddings denote the position of each word in the sequence, which is illustrated in figure 3.

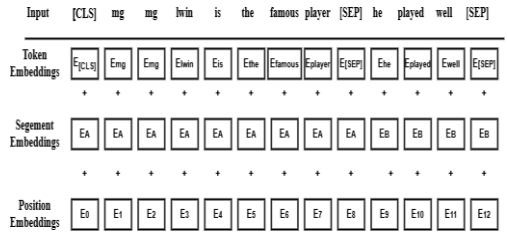


Figure 3. Token Embeddings, Segment Embeddings, Positional Embeddings

3.3.2. BERT Pre-Training

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained deep network developed by google which tries to understand the whole context of a word within a sentence by looking at both its left and right

context. This deep understanding of context is enabled by BERT having been pre-trained on huge corpora such as BooksCorpus (800 million words) and English Wikipedia (2.5 billion words). BERT is based on Transformers [6]. BERT could be read data in both directions, and transformers that assist with understanding context and ambiguity within language. BERT uses only the encoder. There are 12 layers in BERT-Base and 24 layers in BERT-large. The proposed model is only used BER-Base. Each encoder layer consists of Multi-Head Self-attention mechanism, feed-forward neural network and layer normalization and residual connections, which is illustrated in figure 5.

BERT is pre-trained on large text corpora using two unsupervised tasks that are masked language modelling and next sentence prediction.

- **Masked Language Modelling:** Masked language modeling (MLM) is one of the key pretraining tasks used in the BERT (Bidirectional Encoder Representations from Transformers) model. BERT is designed to learn bidirectional context in the sense that it can identify a word based on the surrounding words both preceding and succeeding it. Of the selected tokens, 80% are replaced with the special token [MASK], 10% replaced with a random token, and the remaining 10 % are left unchanged. The model is then trained to predict the original token masked, given the surrounding context. The figure 4 shows the masked language modelling.

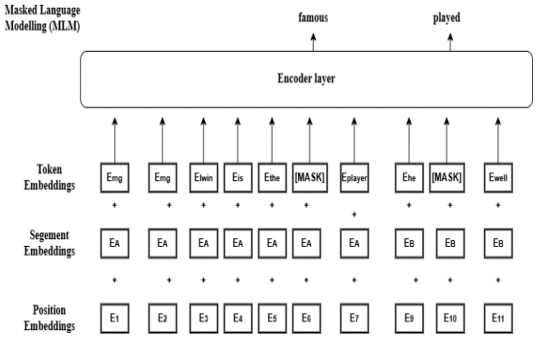


Figure 4. Masked Language Modelling

- **Next Sentence Prediction:** Next sentence prediction is a pre-training task utilized in BERT to allow the model to understand sentence relationship. NSP teaches it to understand the relationship between two sentences. In some cases, the second sentence is the actual sentence that follows the first one in the original text. In another cases, the second sentence is a randomly chosen sentence. BERT is then trained to predict whether the second sentence is actually the actual next sentence or not.

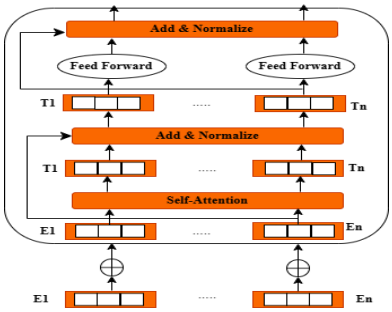


Figure 5. BERT Encoder Layer

3.3.3. Fine-Tuning of BERT

Fine-tuning of BERT consists of taking a pre-trained BERT model and fine-tuning it on particular task, i.e., sentiment analysis, question answering, or text classification. Instead of training a large language model from scratch, fine-tuning allows to take the knowledge of language. Add a small, task-specific layer, classification head on top of the BERT model and train the entire model or just the added layers on labeled data for a few more epochs. The pre-trained BERT model produces contextualized token embeddings, and the special [CLS] token's embeddings is typically used to represent the whole input sequence. The different types of classifier head used in fine-tune BERT model are following:

- **Linear Layer:** A linear layer (also called a dense or fully connected layer) performs a simple linear transformation on its input. It takes a weight matrix and a bias vector and projects the input feature directly onto the input without non-linear activation as calculated as:

$$y = Wh + b \quad (1)$$

Where:
y = output vector from BERT encoder
W = weight matrix
h = output vector
b = bias vector

- **Multi-Layer Perceptron (MLP):** Multi-layer Perceptron typically means feed forward neural network at least one hidden layer. Instead of having a linear layer alone for classification, a multi-layer perceptron consists of one or multiple layers with nonlinear activation functions (such as ReLU) to learn complex patterns in data. A

multi-layer perceptron with two layers can be expresses as:

$$a_1 = ReLU\ (W_1 \,.\, h_{[CLS]} + b_1) \quad (2)$$

$$z = W_2 .\, a_1 + b_2 \quad (3)$$

Where:

$h_{[CLS]}$ = the hidden state of the [CLS] token from BERT

W_1 = Weight matrix for the first layer

W_2 = Weight matrix for the second layer

b_1 = bias vector

b_2 = bias vector

z = final output logits

4. Results and Discussion

4.1. Accuracy Comparison of the Different Amount of Datasets

As Firstly, this system was trained and tested with fewer dataset 610, and the accuracy is 90%. Then, this system was trained and tested with 2500, 5000, 10000, and 21587 documents and the accuracy are 95%, 95%, 96% and 97% respectively. The result show that, the more documents are trained, the better accuracy result is achieved. Table 4 and figure 6 shows the accuracy comparison of different amount of data.

Table 4. Accuracy Comparison of Different Amount of Data

Total Number	Training Dataset	Testing Dataset	Accuracy (%)
610	488	122	90.16
2500	2000	500	94.60
5000	4000	1000	95.0
10000	8000	2000	95.95
21578	17262	4316	97.36

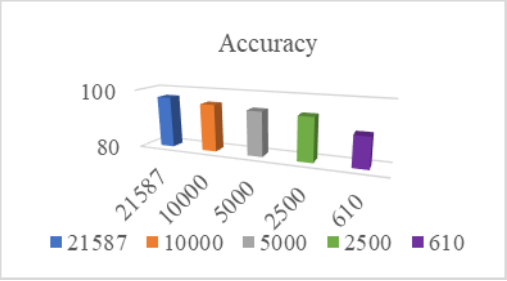


Figure 6. Chart for Accuracy Comparison of Different Data

4.2. Accuracy Comparison with Different Hidden Layers

The proposed model compared with other classification head such as linear layer and multi-layer perceptron during fine-tuning the model. By using linear layer, the accuracy obtained 96% in BBC dataset, 97% in collected dataset, and 97% in hybrid dataset. The accuracy obtained 97% in BBC dataset, 98% in own news dataset, and 98% in hybrid dataset by using multi-layer perceptron with one hidden layer. By using two hidden layers, the system gets the accuracy 97% in BBC dataset, 98% in own collected dataset and 98% in hybrid dataset respectively. The more hidden layers are used, the more accuracy achieve. Table 5 and figure 7 show the different accuracy of using different classification head during fine-tuning.

Table 5. Accuracy Comparison of Different Hidden Layers

	Accuracy		
	Linear layer	Multi-Layer Perceptron	
		One hidden layer	Two hidden layers
BBC Dataset (2225) Documents	95.51	96.63	96.63
Own Dataset (21578) Documents	97.36	98.14	98.24
Hybrid Dataset (23803) Documents	97	98	98.03

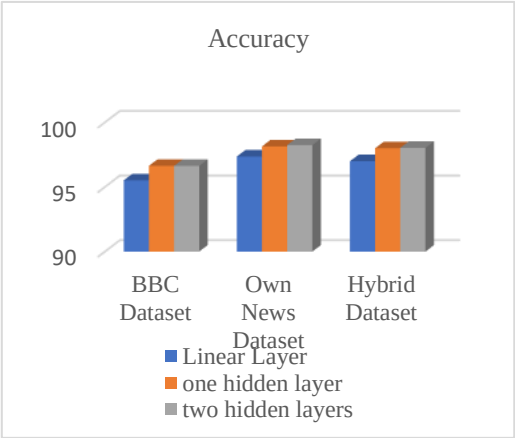


Figure 7. Chart for Accuracy Comparison with Different Hidden Layers

4.3. Accuracy Comparison with Different Optimizers

During training to minimize the loss function and to improve model performance, optimizers are used for adjusting a neural network's weights and bias. The table 6 and figure 8 show the accuracy comparison of optimizers which are Stochastic Gradient Descent with Moment (SGDM), Adaptive Moment Estimation (ADAM), Root Mean Square Propagation (RMSProp).

Table 6. Accuracy Comparison with Different Optimizers

	SGDM	ADAM	RMSProp
BBC Dataset	96.63	95.51	89.21
Collected Dataset	98.03	97.36	98.80
Hybrid Dataset	97.58	97	97.67

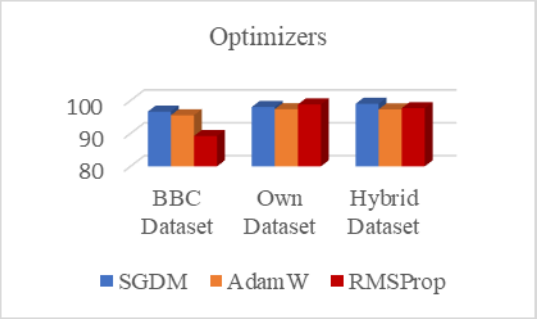


Figure 8. Chart for Accuracy Comparison with Different Optimizers

5. Conclusions

With the increase in data volumes, text classification has now become a necessary tool for organizations. The automated machines allow them to improve their performance with time and save a considerable amount of time and resources compared to manual mechanisms. The proposed system present text classification technique based on deep learning models. The model is trained and tested on BBC dataset having 2225 news articles with five classes. Own news dataset is created and collected the data with additional class "Travel". Moreover, this system trained and tested with hybrid dataset by adding of BBC dataset and collected news datasets. The accuracy of each classifier is 97%, 98% and 99% respectively for collected dataset. The proposed system shows that the more data the system used in training process, the better accuracy result is achieved.

In future, more news data is collected and then put into the dataset, and classify them with other classifier as the further extension. Thus, this system could be used effectively in business to reduce the workload of manual text classification which will save time and effort required in the manual process.

Acknowledgement

First of all, I would like to express my deep gratitude to my parents and all the teachers. It could not have bee possible to complete this work without help from many resources. I would like to express my heartfelt gratitude to my Pro-Rector, Dr. Khin Sandar Aung for giving me a golden opportunity to do this research. I would like say a special thank to my supervisor, Dr. Nan Saw Kalayar. Her support, guidance and overall insights in this an inspiring experience for me. I would like to thank my family for supporting me during the compilation of this work.

References

[1] J. Ahmed and M. Ahmed, "Online news classification using machine learning techniques," *IIUM Engineering Journal*, vol. 22, no. 2, pp. 210–225, Jul 2021.

[2] M. Jannat, E. Hossain, M. M. Hoque, and M. A. Rahaman, "Multi-class Short Text Classification Using Ensemble of Deep Learning Classifier," *Lecture notes in networks and systems*, pp. 478–488, Oct. 2022.

[3] Y. Goel and S. Samantaray, "MULTI-LABEL NEWS CLASSIFICATION USING BI-LSTM 1*," vol. 9, no. 11, pp. 2320–2882, 2021.

[4] O. Blanc, A. Pritzkau, U. Schade, and M. Geierhos, "CODE at CheckThat! 2022: Multi-class fake news detection of news articles with BERT," *ResearchGate*, Aug 2022.

[5] "Devlin et al 2018 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Czlwang.com*, 2018.

[6] X. She and D. Zhang, "Text Classification Based on Hybrid CNN-LSTM Hybrid Model," *IEEE Xplore*, Dec 2018.

[7] Muhammad Abdul-Mageed, A. A. Elmadany, and El, "ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic," Dec 2020.

[8] B. Lutkevich, "What is BERT (Language Model) and How Does It Work?," *SearchEnterpriseAI*, Jan 2020.

[9] Momna Ali Shah, Muhammad Javed Iqbal, N. Noreen, and I. Ahmed, "An Automated Text Document Classification Framework using BERT," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, Jan 2023.

Resume Screening Job Classification System Using Three Machine Learning Algorithms

Nan Saw Kalayar, Chaw Yati Oo
University of Computer Studies (Taunggyi)
saw.kalayar@gmail.com, chawyatioo8@gmail.com

Abstract

The lack of support, direction, and tools in many underdeveloped countries which makes it difficult for students to choose the most appropriate professional path. Many students especially most fresh graduated students are unclear about their goals and unsure which career path best fits with their skills and interests. A Resume screening job classification system is suggested to solve this problem by using Machine Learning algorithms. In this system, data has been gathered especially for Information Technology (IT) fields to aid IT students in Myanmar. There are 1000 Curriculum Vitae (CV) data collected from the students of the University of Computer Studies (Taunggyi) for the dataset. Among them, 80% is used for training and 20% is used for testing. The data is cleaned in the preprocessing step, and Term Frequency-Inverse Document Frequency (TF-IDF) is used for feature extraction and vectorization. After that Training the model using three machine learning algorithms. And then classification process is done and their accuracy are compared. Random Forest classifier gains the highest accuracy of 99.09% among three classifiers on the 1000 CV dataset. The automatic resume screening system could help students identify suitable jobs and careers that match their skills and knowledge.

1. Introduction

These days, there are a lot of different jobs, complex skills and requirements; finding a suitable job is the most difficult for employees. Especially for freshly graduated students, it is very difficult to get the right job due to the lack of job market, and also their skills and experiences are not compatible with the job requirements. A recommendation system is a kind of artificial intelligence (AI) tool that processes user data and offers recommendations depending on preferences, behavior, or users' interactions. It is used in several fields like entertainment, healthcare, e-commerce, and job recruiting.

In the job market, there are a lot of job recommendation systems that recommend the most suitable job. However, most of the online career advice systems are not particularly for the computer science field and some career suggestion systems could suggest a job not given in Myanmar as some systems are being created by foreign developers. Moreover, many current systems depend mostly on personality or interest-based exams; students' real abilities, academic history, or the local labor market are not be analyzed. The realistic, skill-based employment recommendations are provided in this system for IT students in Myanmar to overcome these constraints.

Analyzing a resume is the main process of resume screening to provide for various kinds of employment [4]. Education, experience, skills, and any other information that may show on the CV help to determine that the student is the best fit with the requirements. After processing the data, the recommendation algorithm gives suggestion for related job roles that match with students' profiles.

2. Relate Work

R. Narula, V. Kumar, R. Arora, and R. Bhatia presented a system for applying recommendation of job by using machine learning and Natural Language Processing techniques. This is a hybrid system by uses collaborative and content-based filtering. Word2Vec and cosine similarity are the system's key features. The model extracts the features like education, skills, and experience. The system has good effectiveness in matching jobs to the right candidates with 0.47 precision, 0.59 F1-score, recall, and 0.86 cosine similarity score for matching suitable jobs. [1].

S. Chandraghandi, P. Anamika, S. Shilpa, N. Santhoshsivan, and R. Kamalakkannan provided a system of resume screening with the use of Natural Language Processing (NLP) techniques and TF-IDF for reducing the need to choose candidates manually. Cosine similarity to job postings by matching the job description with user skill sets. Exploratory Data Analysis (EDA) is applied for data cleaning and preprocessing, and the overlap coefficient is used to increase accuracy [2].

T. Natschlager, B. Freudenthaler, and J. Martinez-Gil described a job recommendation system using Random Forests (RF) and Support Vector Machines (SVM) classifiers for applying suitable jobs for employers. In this approach, information like location, salary, and company size is retrieved for analyzing user requirements. The two methods get better performance than non-learning baselines, in which Random Forest obtains higher performance for tasks with complex interactions [3].

Y. Li, X. Zong, and K. Feng suggested that the resume screening approach use machine learning techniques and a dynamic talent model for corporate Internet recruitment. This system uses techniques like SVM, decision trees, and Web crawlers, and Baidu Paddle is used for data collection. As an experimental result, this approach helps in increasing resume efficiency rates above 60% than the rates of manual screening, which is only 20 % [5].

S. R. Dhotre and R. S. Naik presented a machine learning-based resume suggestion system to simplify the hiring process. Naive Bayes algorithm and K-Nearest Neighbor (K-NN) algorithms is used to classify the data after making the preprocessing steps like tokenization and TF-IDF feature extraction. Cosine similarity is used to match resumes and job descriptions. Experimental findings show the model's efficiency is higher when working with resumes from a huge pool. The challenges of this approach are resume formats with no standard and manual screening inefficiencies [6].

3. Methodology

3.1. Proposed System Design

This system introduced the resume screening job recommendation system by using three different classifiers, where user can upload their CV to obtain the recommendation. At the first steps, the Own CV dataset is created by collecting the CV documents from students and then the CV data is loaded from the dataset. After extracting the data, the next step is the data preprocessing steps, which preprocess the data to obtain high clarity and accuracy for training the dataset. In this stage, data cleaning, stopwords removal, Tokenization, Lemmatization, and TF-IDF vectorization are performed. Then the cleaned data is split into a training dataset and a testing dataset. After splitting the dataset, the data is processed by using classification, and then the recommended category is delivered to the user.

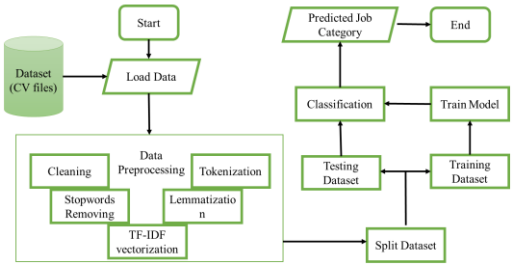


Figure 1. System flow of job recommendation

3.2. Own CV Dataset

The dataset includes 10 categories and total of 1000 CV documents, which are collected from the students of the University of Computer Studies (Taunggyi). By comparing the gpt_dataset, this dataset contains more than 2 categories. These data are especially gathered for the Information Technology (IT) fields to support the IT students of Myanmar. The dataset contains two features: Resume and Category in which the target feature is Category.

Table 1. Category of own dataset

Number	Category
1	Human Resource (HR) Specialist
2	Accountant
3	UI/UX Designer
4	Web Developer
5	Application Developer
6	Data Scientist
7	Network Engineer
8	AI Engineer
9	Database Administrator (DBA)
10	Cyber Security Specialist

In the following figure, 80 % of the dataset, which is 800 CV files, are used for the training

phase, and 200 CVs, which are 20 % of the dataset used for the testing phase.

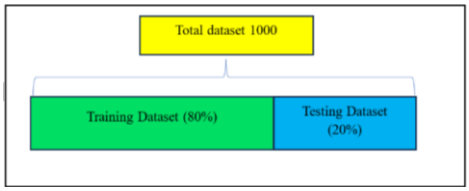


Figure 2. Train-test split

3.3. Gpt_datset

The gpt_dataset, which is taken from the Kaggle website, includes 400 CVs and 8 job categories [10]. From the gpt_dataset, some job categories such as Backend Developer, Frontend Developer, and Full Stack Developer as Web Developer category, Machine Learning Engineer as Data Scientist category, and Mobile App Developer as Application Developer category were selected. This selected portion of the GPT dataset was combined with own CV dataset to form a hybrid dataset. Then, the obtained hybrid dataset is trained by using three different classifiers to support the effective resume classification and recommendation system.

Table 2. Category of gpt_dataset

Number	Category
1	Backend Developer
2	Cloud Engineer
3	Frontend Developer
4	Data Scientist
5	Full Stack Developer
6	Python Developer
7	Mobile App Developer
8	Machine Learning Engineer

3.4. Term Frequency-Inverse Document Frequency (TF-IDF)

A numerical statistic called TF-IDF shows the relevance of a word inside a document to a corpus, or group of documents [8]. It can transform the textual data into numerical feature vectors employing identifying and prioritization the most relevant phrases and by means of de-emphasizing the weight of common words or shorter words. In students' CVs, TF-IDF shows important technical words like "Python," "Java," or "Machine Learning," so they help to make the decision between job categories.

Table 3. Example documents

Number	Document
Doc 1	Bachelor skills Java Java
Doc 2	skills html Java html

The table above displays example documents forming the corpus used for the TF-IDF computation.

3.4.1. Term Frequency (TF)

TF counts how often a word appears in a document. It is calculated as the proportion of the occurrence of the term in a document to the overall number of terms in that text [8].

$$TF = \frac{number\ of\ times\ the\ term\ appers\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document} \quad (1)$$

Table 4. TF calculation for document1

Terms in Document 1	Number of Terms	TF calculation for document 1
Bachelor	1	1/6=0.1667
skills	1	1/6=0.1667
Java	2	2/6=0.333

Table 5. TF calculation for document2

Terms in Document 2	Number of Terms	TF calculation for Document 2
skills	1	1/6=0.1667
Java	1	1/6=0.1667
html	1	2/6=0.333

The above tables illustrate the TF values of each term in Document 1 and Document 2.

3.4.2. Inverse Document Frequency (IDF)

IDF measures the rarity of a term across a collection of documents. It is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term [8].

$$IDF = log\left(\frac{total\ number\ of\ documents\ in\ the\ corpus}{number\ of\ documents\ that\ contain\ the\ term}\right) (2)$$

Table 6. IDF calculation for document 1 and 2

Term	Frequency over 2 Docs	IDF
		Doc. 1 & Doc. 2
Bachelor	1	$\log\left(\frac{2}{1}\right) = 1$

skills	2	$\log\left(\frac{2}{2}\right)= 0$
Java	2	$\log\left(\frac{2}{2}\right)= 0$
html	1	$\log\left(\frac{2}{1}\right)= 1$

The above table shows the calculation of IDF values for the whole corpus made up of document 1 and document 2.

3.4.3. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is the combination of TF and IDF ratings. It is applied in information retrieval and text mining to determine how relevant a word is to a document in a collection or corpus [8].

$$TF= TF \times IDF \quad (3)$$

Table 7. TF-IDF calculation for document 1

Terms	TF	IDF	(TF*IDF)
Bachelor	0.1667	1	0.1667
skills	0.1667	0	0
Java	0.333	0	0

Table 8. TF-IDF calculation for document 2

Terms	TF	IDF	(TF* IDF)
skills	0.1667	0	0
html	0.333	1	0.333
Java	0.1667	0	0

The above tables present the calculation of TF-IDF values for document 1 and document 2. When calculating the importance of words in two documents using TF-IDF, Bachelor is an important word for document 1, and html is an important word for document 2.

3.5. Random Forest Classifier (RF)

Random Forest, a supervised machine learning technique, is based on ensemble learning [7]. It can be used in two versions, one for problems with regression and one for problems with classification. Random forest is one of the most flexible and easy-to-use algorithms. Based on the provided data samples, many decision trees are built in Random Forest. Each tree provides an estimation, and voting is then used to find the best answer.

The following describes the workflow of the Random Forest algorithm.

- K samples are selected at random with replacement from the training dataset.
- From the data sample, which is chosen randomly, a decision tree is built.
- Repeat steps 1 and 2 for a set number of times (n).
- The final prediction or final result for regression assignments is the sum of all the predictions made by each decision tree.

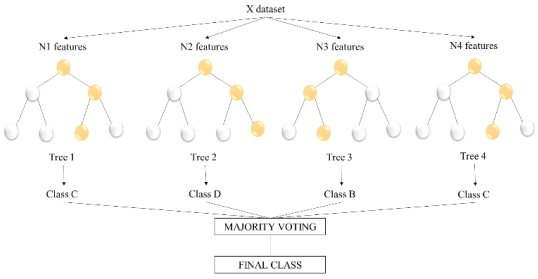


Figure 3. System flow of the random forest classifier

In decision trees in Random Forest, metrics like Gini impurity, information gain, or mean square error (MSE) can be used to determine the quality of the split. This recommendation system applies information gain to deliver recommendations.

3.6. Decision Tree (DT)

In machine learning, Decision Trees (DT) are special because they are simple, easy to understand, and flexible [7]. It starts with a root node that represents a sample, and the root node then divides the population or sample into two or more equal groups using a technique called splitting. Then the sub-nodes split again into another group. The node that is not separated is defined as a leaf or terminal node. For a job recommendation system, a decision tree can analyze the data of students' CV files, things like their skills, experience, and qualifications, to provide the best job category for each student.

3.7. K-Nearest Neighbors (KNN)

A simple, non-parametric, and lazy learning algorithm called K-Nearest Neighbors (KNN) is

used for jobs like classification and regression [9]. It is based on the idea that things that are similar are close to each other. The data points are classified in KNN based on how the groups around them are sorted. With a job recommendation system for IT students, the K-Nearest Neighbors algorithm matches a student's resume with similar resumes from the dataset to find the best job. KNN looks for the most similar resumes, known as "k" based on these factors when someone enters a new CV. It suggests a job based on the group that most of those neighbors fall into.

4. Results and Discussion

4.1. Accuracy Comparison of Three Datasets

The following table shows a comparison of the accuracy of three datasets: the Own job dataset, the gpt_dataset, and the hybrid dataset. The accuracy of Random Forest is obtained better than the others (99.13 %) in Hybrid Job dataset which is combined own CV dataset and gpt_dataset. It is the most accurate model across all types of datasets and which makes it the most suitable for a job recommendation system.

Table 9. Accuracy comparison of three datasets

Dataset	Accuracy		
	RF(%)	DT(%)	KNN(%)
Own CV Dataset	99.09	99.08	97.59
gpt_dataset	98.91	83.85	60.76
Hybrid Dataset	99.13	99.06	97.97

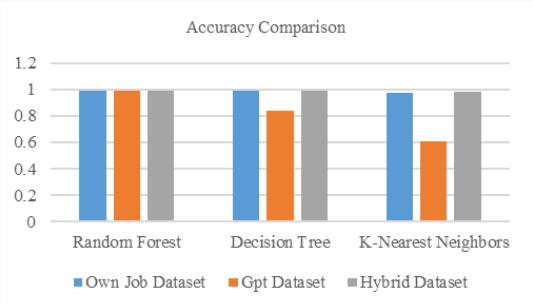


Figure 4. Accuracy comparison result of three datasets

The above figure presents the results of accuracy by comparing three different datasets.

4.2. Accuracy Comparison of Different Amounts of Data

The accuracy of training in three classification models is presented in the following table. The different amount of data, which are 100, 150, 300, 500, 700, and 1000 documents are used for training. By comparing the accuracy of three classifiers, the random forest obtains the highest accuracy. The larger data is used, the higher accuracy is achieved.

Table 10. Accuracy comparison of different amounts of data

Data	Accuracy		
	RF(%)	DT(%)	KNN(%)
100	80	55	85
150	90.63	87.5	96.88
300	98.33	96.67	94.16
500	99.22	98.44	95.07
700	98.07	97.91	94.79
1000	99.09	99.08	97.59

The following figure illustrates accuracy comparison of three different datasets in different amount of data.

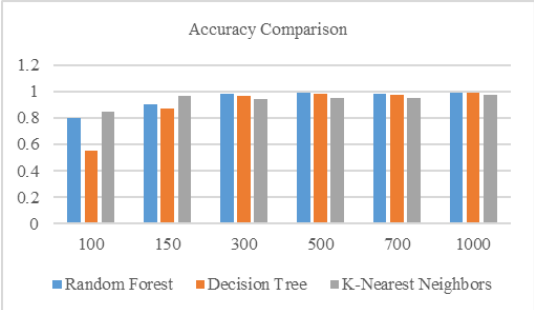


Figure 5. Accuracy comparison result of different amount of data

4.3. Precision, Recall, F1-score comparison of Three Datasets

The following table presents the results of precision, recall, and F1-score in three different classifiers on 1000 CV documents, in which the highest weighted average scores are obtained by the Random Forest classifier.

Table 11. Precision, Recall, F1-score comparison of three classifiers

Classifier	Precision	Recall	F1-score
RF	0.99	0.99	0.99
DT	0.99	0.98	0.98
KNN	0.98	0.98	0.98

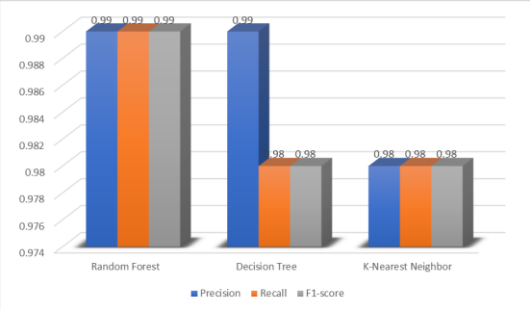


Figure 6. Precision, Recall, F1-score comparison of three classifiers

The above figure presents the result of precision, recall and f1-score comparison.

5. Conclusions

This system provides a job recommendation system based on resume screening by using machine learning techniques to help graduates with their talents, education, and experience find appropriate career choices. The suggested model can properly fit applicants to job openings in the computer science sector using TF-IDF extraction of essential elements from CVs and system training on a dataset of 1,000 CVs. This system offers high accuracy and can be a useful tool for students and colleges trying to enhance recommendations of job results by the use of preprocessing, vectorization, and classification. Future improvements could consist of adding more sophisticated NLP methods, expanding the dataset, and combining real-time job market data to perform a more accurate job recommendations system with higher relevance and correctness.

Acknowledgement

First, I want to thank my parents and all of my teachers from the bottom of my heart. This paper could not have been finished without a lot of different sources of help. I want to thank our Pro-Rector, Dr. Khin Sandar Aung, for giving me the chance to do this study. Then I want to thank our professor, Dr. Nang Kaythi Hlaing, who helps me with my studies. Please accept my sincere thanks to Dr. Nan Saw Kalayar, who is my supervisor. Her help, advice, and general thoughts made this an inspiring experience for me. Also, I want to thank all the students from the University of Computer Studies (Taunggyi), who gave me records. I would like to thank my family for being there for me while I worked on this work.

References

[1] R. Narula, V. Kumar, R. Arora, and R. Bhatia, “Enhancing Job Recommendations Using NLP and Machine Learning Techniques,” TIJER - International Research Journal, vol. 10, no. 10, October. 2023, pp. a347–a356.

[2] S. Chandraghandi, S. Shilpa, P. Anamika, R. Kamalakkannan, and N. Santhoshsivan, “Resume screening using TF-IDF,” Int. J. Adv. Res. Comput. Commun. Eng., vol. 11, no. 5, May 2022, pp. 720–731.

[3] J. Martinez-Gil, B. Freudenthaler, and T. Natschläger, “Recommendation of job offers using Random Forests and Support Vector Machines,” Workshop Proceedings of the EDBT/ICDT 2018 Joint Conference, Vienna, Austria, March. 2018, pp. 22–27.

[4] M.N.V.S. Raghavendra, “Resume Screening Using Machine Learning”, Journal of Engineering Sciences, JES Publications, Visakhapatnam, India, 2022, pp. 402–407.

[5] X. Zong, Y. Li, and K. Feng, “Research on resume screening methods in corporate Internet recruitment based on machine learning,” Proc. SPIE, vol. 13105, 131053R, April. 2024.

[6] R. S. Naik and S. R. Dhotre, “Resume recommendation using machine learning,” Int. J. Creative Research Thoughts (IJCRT), vol. 10, no. 7, July 2022, pp. 784–792.

[7] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, “Random Forests and Decision Trees”, IJCSI International Journal of Computer Science Issues, IJCSI, Peshawar, Pakistan, September 2012, pp. 272-278.

[8] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents”, International Journal of Computer Applications, IJCA, Sintok, Kedah, Malaysia, July 2018, pp. 25-29.

[9] S K. Taunk, S. De, S. Verma, and A. Swetapadma, “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification”, Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019), IEEE, Bhubaneswar, India, 2019, pp. 1255–1260.

[10] J. Jagadeesh, “Resume Dataset”, Kaggle, Online Dataset, 2020.

Text Meets Ensemble Learning: Prediction House Prices with TF/IDF and XGBoost

Khine Cherry Htun, Yin Nyein Aye
University of Computer Studies (Taunggyi)
Khinecherryhtun@ucstgi.edu.mm , yinnyeinaye@ucstgi.edu.mm

Abstract

House Price Prediction is an important problem that could benefit both buyers and sellers. A lot of house price prediction model solely based on numeric features. Always neglect the important of house description text or house listing that are mentioned. This paper will focus to compare the performance of three factors: when only using house description, only using numeric description and combine both text and numeric data. TF/IDF is used for Text description and XGBoost algorithm is used for Prediction. The model's prediction accuracy improves significantly when using both numeric and text data together compared to using only numeric or only text data.

1. Introduction

Owning a house is not just a financial investment, it is about personal accomplishment and for many people it is like a lifelong dream. Home or houses are the place where individuals or people find comfort or create memories. The real estate market is critical to the economy as houses or property play important role for buyers, seller and investors. Predicting house price can help investor to make informed decision. Knowing the price range allows buyers to prepare for further purchases and sellers to price their properties fairly.

For many people, buying a house mean achieving a dream or achieving a comfortable lifestyle. This dream can lead to a bad problem if the property is overpriced or undervalued. Overpaying for a house can overload buyers, while selling a house below the actual value can negatively affect sellers. Sometimes the involvement of agents can lead to price inflection. In Shan state Myanmar, cities like Taunggyi, Kalaw, Aungban and Nyaungshwe have become highly desirable for property ownership in the recent years. Taunggyi, the state capital attracts people because of its nature and living conditions. Kalaw, with its cool climate and natural beauty attract people to buy house for staycations. Aungban strategic position as a transport hub appeals to both residents and businesses. Nyaungshwe near the Inle lake, draw attention from tourists and investors. These cities offer unique opportunities, making them ideal for studying trends and prices.

House Price Prediction model often relies on numeric data (number of bedrooms, number of bathroom and etc.). Because of that people often negligence on property listings such as (well maintain, recently renovated and etc.) However, with the advent of advanced machine learning techniques and the increasing availability of property descriptions, new opportunities have arisen to enhance prediction accuracy by incorporating text-based data.

This paper aims to develop house price prediction system that contains textual data to enhance the house price prediction. The textual data contain such as such as property condition, amenities, and neighborhoods descriptions offer a more comprehensive view of factors influencing prices. By combining methods, the TD/IDF for text vectorization and XGBoost for prediction, this model can make informed decisions and provide users a more understanding of price estimates.

2. Related Work

In the case study [1] provides insights into Melbourne Housing Market. The data reduction is carried out by stepwise and PCA (Principal Component Analysis) techniques. Different Methods (Linear regression, neural network) is used for evolution. In [2] estimated the value of land in Andhra Pradesh with eight different factors (Security, Markets, schools, collages, hospitals, Registration value Environment and pollution).

Used Multiple linear regression and used two scatter plots to show positive impact on the land value. In [3] predict the house price Karachi city, Pakistan using XGBoost and got 98% accuracy. Based on textual data in [4] compare three scenarios (textual data, numeric data and both numeric and textual data) are used to predict the house price. Three techniques for word embedding (TF/IDF, Word2Vector and BERT) and Four regression algorithm (Random Forest, XGBoost, SVM, and Deep Neural Network are used to compared. In [5] used TF/IDF for text Pre-processing and create 845 new unique textual features. Used the RFECV (the recursive features elimination with Cross Validation) to reduce the textual features from 845 to 38 which are the most significant ones. Used LightGBM model for prediction and SHAP (Shapely Additive Explanations) Plot to show positive and negative values. In [6] combined both structural features (such as number of bathroom and bedroom and a vectorized (textual description). Used four machine learning algorithms (extremely randomized trees, Gradient boosting, random forest, XGBoost and LightGBM). And for word embedding used TF/IDF and Bow (bag of words). Used two datasets to get insights of the text features like which words effect the price and the words that extract from both dataset which effect the price are not the same.

3. System Architecture

In first phase, the data collection, gathering relevant information such as number of bedrooms, bathrooms, property size, location, property type and property descriptions. For text data, data pre-processing is applied and TF/IDF is used to convert the property description into a numerical format, making them suitable for prediction. Similarly, the numeric data (such as bedrooms, bedrooms and property size) and categorical data (property type and location) is pre-processed through normalization and encoding. Once the data is prepared, it is split into training (80%) and testing test (20%). The training data is used to trained an XGBoost Model which learns relationships between features and house prices. After training, the model is test on the test set to evaluate its performance. The system performance is evaluated using metrics like R-squared, MSE and RMSE. Figure 1. Show the system architecture.

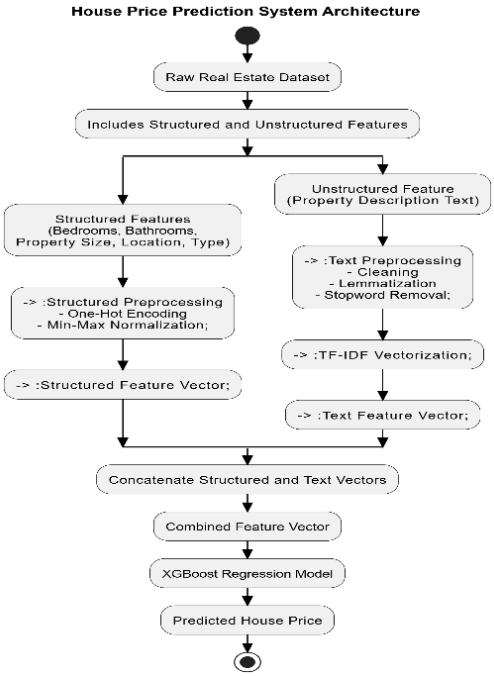


Figure 1. System Architecture

4. Methodology

This study, data is collected from various real estate websites (I Myanmar house, Shwe Property, Property .com and Facebook). The dataset contains 1500 data and 7 attributes including target features price. The dataset primarily focuses on properties in Taunggyi, Kalaw, Aungban and Nyaungshwe. Table 1. shows the data obtain from the websites.

Table 1. Data description

No	Features	Description	Example	Data Type
1	Property Type	The type of property mentioned. categorical data	Apartment, House	Categorical Data
2	Location	Information about the location of the house.	Taunggyi, Kalaw, Aung Ban Nyaungshwe,	Categorical

3	Bedrooms	Number of Bedrooms of the property	3bedrooms,2 bedrooms	Numeric Data
4	Bathrooms	Number of bathrooms of the property.	1bathroom.,2bathrooms	Numeric Data
5	Property size	Sizes of the property	1280 square feet	Numeric Data
6	Description	Textual description detailing the property.	Good location and very valuable. wide land, flat land, land registration form 105, owner's name. Constructed. All prepared	Textual
7	Price	The selling price of the property	1,00000000 MMK	Target Value

4.1. Data Pre-processing

The collect data has two main features: the standard non-textual description features and the textual description data.

4.1.1. Non-textual Data

For example, there are two four numerical features and two categorical. Pre-processing techniques are applied.

- One hot encoder: Two categorical data are represented which are Locations and Property type. For better performance convert both categorical data into numeric data using one hot encoding. For instance, last location has four categories (Taunggyi, Kalaw, Aungban and Nyaungshwe) Taunggyi become 1 when it is presented and one hot encoding same goes with other categories. And property types have two -House become 0 and apartment become 1.[4]
- Numerical Feature Normalization: The numerical feature has different value ranges, convert the feature to same range. Bedrooms, bathrooms property types and price are normalized by using MinMaxScaler the price is normalized to scale between 0 and 1.[4] The Min-Max Scaler is defined by the following formula:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

- Where Here, X is the original value, X_{min} is the minimum value,X_{max} is the maximum value, X_{scaled} is the scaled value. For example, MMK 400,000,000.00, MMK 560,000,000.00 and MMK 2,950,000,000.00 transformed using min-max scaler.
- Min-Max scaling for 2,950,000,000.00=1
- Min-Max scaling for 400,000,000.00 = 0
- Min-Max scaling for 560,000,000.00 =0.67

4.1.2. Data Pre-processing of textual description data for TF/IDF

TF/IDF requires to clean textual data. Thus, why before adapting TF/IDF data pre-processing of textual description is applied.
Abbreviation: Abbreviations are converted to full word so that only one single format is need for TF/IDF algorithm.

- Punctuation: Punctation marks and special characters are removed. For instance, "@, #, and ……"
- Lemmatization: Lemmatization used for the process to transform data to single form. By transforming the words to it base form.
- Stop words: Need to remove stop words. There are two types: English stop words and real estate stop words. Real estate stop word are common in real estate such as "area" and "location". The real Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined

before or immediately following the equation. Use estate stop word that are taken from [4] as mention.

- Tokenization: To tokenize the words in the textual description data. The tokens are separated by a space.

Table 2. Preprocess Documents

Documents Before Preprocessing	Documents After Preprocessing
Good location and very valuable, wide land, flat land, land registration form 105, owner's name. Constructed. All prepared	good, location, valuable, wide, land, flat, land, registration, form, owner, name, construct, prepare
A large courtyard and located in a good location. A two-stories building in the yard. In addition to the quiet neighborhood and near the Supermarket, school and hospital.	Large, courtyard, locate, good, location, two, story, building, yard, quiet, neighborhood, near, supermarket, school, hospital.
The location is good and very suitable for commercial use. Constructed. All prepared	location, good, suitable, commercial, use, construct, prepare.

4.2. TF/IDF (Term Frequency / Inverse Document Frequency)

TF/IDF is defined as how relevant a word in a document or corpus. Term Frequency (TF) defined as the number of times a term appear in a document. Inverse Term frequency (IDF) measures of how important a term is.[7] TF/IDF is defined as:

$$TF(t,d) = \frac{Frequency\ term\ \text{"t" in document d}}{total\ term\ in\ document\ "d"} \quad (2)$$

$$IDF = log\ \frac{total\ number\ of\ documents}{total\ document\ with\ term\ "t"} \quad (3)$$

For example, there are three documents After all the Pre-processing:

Document 1: good, valuable, wide, land, flat, registration, form, 105, owner, name, constructed, prepared

Document 2: Large, courtyard, locate, good, location, two, story, building, yard, quiet, neighborhood, near, supermarket, school, hospital.

Document3: location, good, suitable, commercial, use, construct, prepare.

TF (good)= 1 / 12 =0.0833

IDF (good)=log (3/1) =0.477

TF/IDF=0.833*0.477=0.0397

4.3. (Extreme Gradient Boosting) XGBoost

XGBoost (Extreme Gradient Boosting) is a machine learning algorithm that come under ensemble learning. XGBoost is used for both regression and classification. XGBoost builds a predictive model by combining the prediction of multiple individual models. The XGBoost works by sequentially. The XGBoost works by sequentially adding weak learners to ensemble with each new learner focusing on correcting error made by existing ones.[8] XGBoost work as the step as like this:

- Calculated residuals.
- Calculated similarity score using the following formula:

$$Similarity\ Score = \frac{\Sigma(\text{Red})^2}{Number\ of\ \text{Red}+\lambda} \quad (4)$$

- Using similarity score calculated Information gain. By calculating information gain can compared trees. The information gain can calculate as:

$$Gain = lsim + rsim - Rsim \quad (5)$$

  - Whereas,
  - lsim =left similarity
  - rsim =right similarity
  - Rsim =Root similarity

- Prune the tree. By calculated difference between gain and gamma (user defined tree parameter) [9]

$$Gain - Gamma \quad (6)$$

If the result is positive do not prune the tree.

- Calculated Output

$$Output = \frac{Sum\ of\ residuls}{No.of\ residuals+Lambda} \quad (7)$$

5. Results and Discussion

In this study, the performance of XGBoost model was valued using three factors: numeric only features, textual description and combination of both numeric and text features. The evaluation

metrics used were R square, spacing. RMSE (root mean square error) and MSE (mean square error). The Results are shown in table 3.

The model that used only numeric features got an R square of 0.5487 showing that the variance of 54.87% in the house price could be explained by numeric data alone. The model has highest error values with MSE of 2.1594 and RMSE of 1.4535, which show a limited predictive accuracy.

When using text only features, the model's improved with the R square to 0.7119 and the MSE and RMSE to 1.3787 and 1.1559. This showed that textual description contains a greater view of properties which numeric data alone cannot.

The combination of both numeric and textual features shows the best performance. The model showed Highest R square of 0.7824 which means that 78.24% variance in price and also have the lowest MSE and RMSE. Figure 2. showed the features set using R square result and figure3. showed that Features set using RMSE result and Figure 4 showed MSE Result.

Based on the comparison of three features set, the best result is obtained using combination of both numeric and text features. Accordingly, a user-Friendly interface is developed for a house price prediction using both structured inputs such as location, number of bedrooms and bathrooms, property size and property type and unstructured inputs, the property description. Figure 5 showed the user interface and Figure 6 showed a Predicted house price.

Table 3. Results Metrics

Parameter	Numeric	Text	Combined (Text+ numeric)
Rsquare	0.5487	0.7119	0.7824
MSE	2.1594	1.3787	1.0411
RMSE	1.4535	1.1559	1.0523

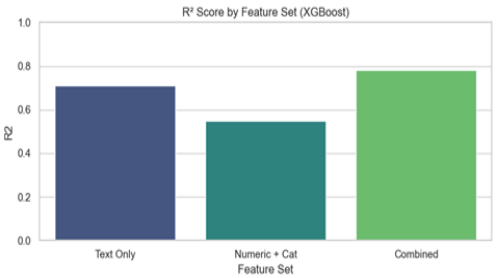


Figure 2. Rsquare Result

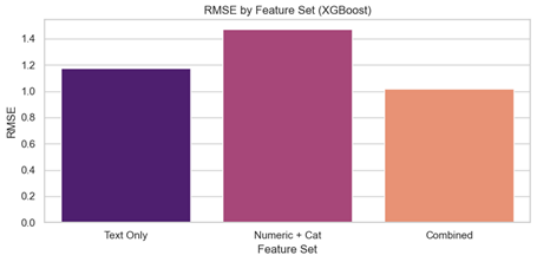


Figure 3. RMSE Result

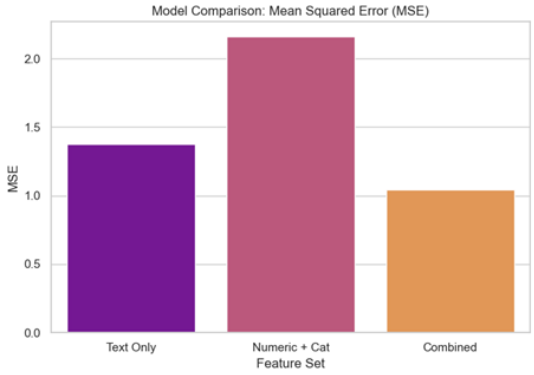


Figure 4. MSE Result

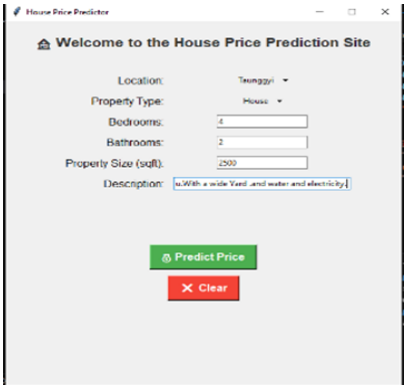


Figure 5. User Interface

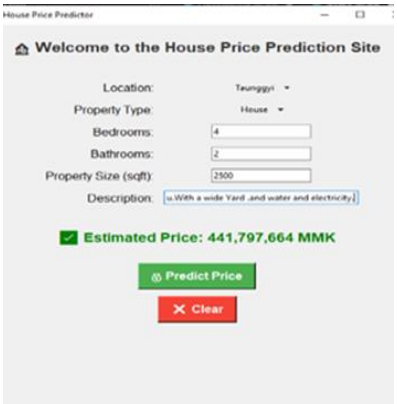


Figure 6. Predicted Price

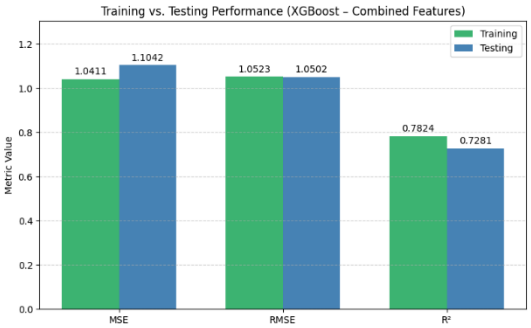


Figure 7. Testing Vs Training Performance

In Figure 7. presents a comparison of the training and testing performance of XGBoost using combined features sets which includes both numeric and text data. The model performs slightly better results on training data across all three metrics (MSE, RMSE and R square) while still maintaining a strong performance on the testing set. The small gap that has been occurred between training and testing show that model is performed well and the model is not overfitting.

6. Conclusion

In this paper, experiments were conducted to evaluate the impact of the text data on house price prediction. The best results were achieved when using text features with a R-squared value of 0.7824. The implementation of text features demonstrated significant improvements in prediction accuracy. In the future, the system will be developed by using Long-short term memory. Word2Vec and BERT for text pre-processing. Additionally, the focus will expand to larger dataset from different cities in Myanmar to improve the system' effectiveness.

References

[1]T. D. Phan, "Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia," 2018 International Conference on Machine Learning and Data Engineering (iCMLDE), Dec. 2018, doi: https://doi.org/10.1109/icmlde.2018.00017

[2] M. Basha, B Ankaiah, J Srivani, and U Dadakalander, "Real Estate Analytics with Respect To Andhra Pradesh: Machine Learning Algorithm Using R-Programming," International Journal of Scientific & Technology Research, vol. 9, no. 4, pp. 2140–2144, Apr.2020,Available: https://www.researchgate.net/publication/348160718_Real_Estate_Analytics_With_Respect_To_Anda_Pradesh_Machine_Learning_Algorithm_Using_R-_Programming

[3]M. Ahtesham, N. Z. Bawany, and K. Fatima, "House Price Prediction using Machine Learning Algorithm - The Case of Karachi City, Pakistan," IEEE Xplore, Nov. 01, 2020. https://ieeexplore.ieee.org/document/9300074

[4]H. Zhang, Y. Li, and P. Branco, "Describe the house and I will tell you the price: House price prediction with textual description data," Natural Language Engineering, pp. 1–35, Jul. 2023, doi: https://doi.org/10.1017/s1351324923000360

[5] Sergey Bushuyev, D. Bushuiev, N. I. Poletaev, Mykola Malaksiano, and D. Kravtsov, "A machine learning method for real estate operation projects forecasting ⋆," CEUR Workshop Proceedings, 2024. 5th International Workshop IT Project Management, ITPM 2024., Dec. 2024, Available: https://www.researchgate.net/publication/387318384_A_machine_learning_method_for_real_estate_operation_projects_forecasting

[6] Luís Fernando Bittencourt, O. Parraga, D. D. Ruiz, I. H. Manssour, and R. C. Barros, "Leveraging Textual Descriptions for House Price Valuation," Lecture Notes in Computer Science, pp. 355–369, Nov. 2022, doi: https://doi.org/10.1007/978-3-031-21686-2_25

[7] "(PDF) Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," ResearchGate. https://www.researchgate.net/publication/326425709_Text_Mining_Use_of_TF-IDF_to_Examine_the_Relevance_of_Words_to_Documents

[8] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A Comparative Analysis of XGBoost," ResearchGate, Nov. 2019, doi: https://doi.org/10.48550/arXiv.1911.01914.\

[9] "Explain the step-by-step implementation of XGBoost Algorithm." https://www.csias.in/explain-the-step-by-step-implementation-of-xgboost-algorithm/

Student Dropout Prediction System Using Three Machine Learning Algorithms

Su Myat Thwe
University of Computer Studies (Taunggyi)
sumyatthwe@ucstgi.edu.mm

Abstract

This paper presents a predictive analysis approach for identifying the key factors influencing student dropout using the Naive Bayes classifier. The primary goal is to provide educators and administrators with insights to reduce dropout rates and enhance student retention. The system explores 13 features and 1 class including the target value. These features encompass various demographic, academic, and socio-economic attributes to build an accurate predictive model. The data is collected from the student dropout in the University of Computer Studies (Taunggyi). The student dropout data is from the academic years 2019 to 2024, and a dataset of 500 students was used in this system. First the dataset is loaded and pass the preprocessing step to speed up the system accuracy. After that the Naive Bayes classifier is applied to train the dataset and test the result. The accuracy result is achieved 86%. The accuracy result is compared with other classifier such as Decision Tree and Support Vector Machine (SVM) in this paper. The data used in this study, including student attendance, attempt, and semesters studied, were found to be significant indicators of student dropout risk. These findings support data-driven when making decision-making in educational institutions.

Keywords—Student dropout, Naive Bayes, Decision Tree, Support Vector Machine, predictive analysis, machine learning

1. Introduction

Student dropout is a persistent challenge faced by educational institutions globally. It can significantly impact both individual students and broader educational outcomes. Identifying at-risk students early is crucial for improving retention rates and academic success. Predictive models, such as the Naive Bayes classifier, offer powerful tools for analyzing dropout factors by leveraging historical data. In this system, the proposed system uses student dropout data from the University of Computer Studies (Taunggyi). The data is from 2019–2024 academic year to build and evaluate the predictive model. This dataset includes 13 features and 1 class. These features include academic performance, attendance and socio-economic status. The proposed system applying the Naive Bayes classifier to train the data in the dataset. After that classification process is done, test the data and output the label of the student drop out "yes" or "no". The accuracy result is compared with other classifier such as Decision Tree and Support Vector Machine (SVM) in this paper. Thus, this paper explores its potential in predicting student dropout and identifying critical influencing factors. By understanding these factors, institutions implement targeted interventions to reduce dropout rates and improve student success. This system contributes to the growing body of research in educational data mining, providing practical insights for data-driven decision-making in education.

2. Related Work

M. Vaarma and H. Li used a dataset of 8813 students dropout record. 24% of the student's record were dropped out and 76% were graduated. They used 16 features including the target value. Their proposed system used Cat Boost (CAT), Neural Networks (NN), and Logistic Regression (LR) Algorithm.[1]

A. Meiriza, E. Lestari, P. Putra, A. Monaputri, and D. A. Lestari illustrated and used 268 student records data, divided it into 141 for training data and 127 for testing data. They used Naive bayes classifier. It was stated that 6 features including target value were used. Their system got the accuracy about 97%.[2]

Harwati, R. I. Virdyanawaty, and A. Mansur described the dataset used in their system. It was collected from the years 2003 to 2007 based on the study period of students at the Industrial Engineering Department (Universitas Islam Indonesia). Eleven (11) features were used and

their paper compared two algorithms, Naive Bayes classifier and Support Vector Machine (SVM). The results show that Naïve Bayes achieved 80.67% accuracy and SVM achieved 60%. [3]

S. R. Sariman, H. Ab Jalil, and E. Marlisah, used a dataset consisting of 2,482 student records collected from primary schools in Selangor. The initial dataset contained 22 attributes, which were reduced to 12 features after feature selection using Info Gain Attribute Eval in WEKA. These features included demographics, academic performance, and socioeconomic indicators. The study applied three classification techniques Naïve Bayes, Random Forest, and Decision Tree to build prediction models. Among them, Random Forest achieved the highest accuracy at 79.57%, followed by Decision Tree (78.16%) and Naïve Bayes (71.68%). [4]

E. Yukselturk, S. Ozekes, and Y.K. Türel used a dataset consisting of 189 student records from an online certificate program between 2007 and 2009. The system included 10 features such as gender, age, educational level, previous online experience, occupation, self-efficacy, readiness, prior knowledge, and locus of control. They applied four machine learning methods: k-Nearest Neighbor (k-NN), Decision Tree (DT), Naive Bayes (NB), and Neural Network (NN). Among them, 3-NN achieved the highest accuracy with 87%, followed by DT with 79.7%, NN with 76.8%, and NB with 73.9%. Genetic Algorithm (GA) was used to identify the most important predictive features, which were self-efficacy, online learning readiness, and previous online experience.[5]

L. T. Kunjumon et al. used a dataset from a Bulgarian university and applied decision trees, neural networks, and KNN classifiers, achieving 73.59% accuracy with neural networks. They later proposed a Naive Bayes-based model incorporating behavioral features, which showed improved accuracy. Similarly, E. Yukselturk et al. compared multiple classifiers and found k-NN most effective for dropout prediction. Previous studies like those by Meiriza et al. and Harwati et al. emphasized the effectiveness of Naive Bayes and SVM, achieving over 80% accuracy using student demographic and academic datasets.[6]

3. System Design and Methodology

3.1. System Design

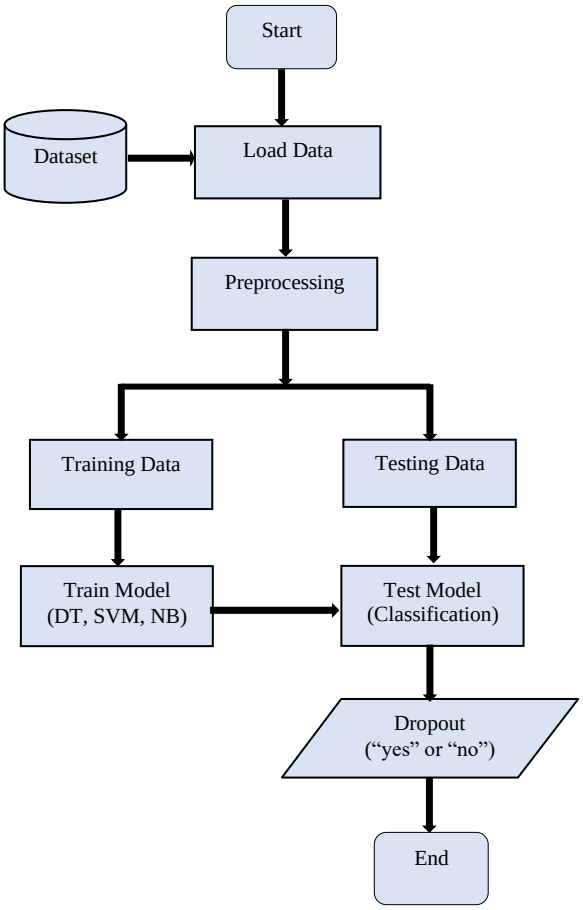


Figure 1. System Design for Student Dropout Prediction using Machine Learning Classifier

This system design outlines a student dropout prediction process using the Naive Bayes classification model. It is beginning with the collection of raw data related to student demographics, academic performance, and other relevant features. The next step is preprocessing, where the data is cleaned, missing values are handled, and categorical features are encoded to make them suitable for model training.

Once preprocessing is complete, the dataset is prepared and then divided into two parts: training data (80%) and testing data (20%). The training data is trained by using the Naive Bayes classifier, Decision Tree and Support Vector Machine (SVM) which learns the statistical patterns and relationships within the input features that contribute to student dropout behavior. After training, the model is evaluated using the testing data. This involves making predictions on unseen data to assess how well the

model generalizes. The prediction results indicate whether a student is likely to dropout “yes” or “no”.

Finally, based on the prediction output, appropriate interventions or actions can be planned in the organization. The process is showing clearly as in Figure 1.

3.2. Dataset

The dataset used for this study consists of 500 student records, covering both academic and demographic information. This system uses 13 features and 1 class to predict student dropout including the target values. These features include demographic, academic, and socio-economic. Among these features, 6 features are numerical data and 8 features are categorical data, which are critical in identifying dropout risk. The 13 features and 1 class are student ID, gender, course, marital status, attendance rate, number of attempts, number of family members, father's qualification, mother's qualification, father's occupation, mother's occupation, semester, and dropout status (“yes” or “no”). The data is divided into training (80%) and testing (20%) sets to evaluate the model’s performance. The features include six numerical attributes and eight categorical attributes, providing a balanced representation of the factors affecting student retention. Table 1 shows the list of 13 features and 1 class used in the student dropout prediction system, including their types as either numerical or categorical.

Table 1. Description of Features

Sr. no.	Feature	Data Type	Description
1.	Id	Numerical	A unique identifier for each student.
2.	Age of Enrollment	Numerical	The age of student when at enrolls time.
3.	Gender	Categorical	Male/ Female
4.	Semester	Numerical	The total semesters.
5.	Course	Categorical	CS/ CT
6.	Marital status	Categorical	single or married.
7.	Attendance	Numerical	percentage
8.	Family Member	Numerical	Total family members of student.
9.	Attempt	Numerical	No. of attempted the course or exams.
10.	Father Qualification	Categorical	Student’s father qualification.
11.	Mother Qualification	Categorical	Student’s mother qualification.
12.	Father Occupation	Categorical	Student’s father occupation.
13.	Mother Occupation	Categorical	Student’s mother occupation.
14.	Dropout	Categorical	Yes/No

3.3. Preprocessing

Preprocessing is a critical step in preparing data for machine learning models. In this system, the dataset consist 500 student records with 13 features and 1 class including academic, demographic, and socio-economic data. To ensure accurate predictions using the Naive Bayes classifier, several preprocessing steps are applied. Firstly, missing values in categorical fields are filled using the mode value, which is the most frequently occurring value. For example, if 'Marital Status' has missing data, they are replaced with the most common value such as 'Single’. The missing of numerical values is put by the mean of that features. Next, categorical data is transformed into numerical format using encoding techniques. Label Encoding assigns each category a unique integer—for instance, 'Male' becomes 0 and 'Female' becomes 1 in the 'Gender' column. One-Hot Encoding converts categories like 'Course' (e.g., 'CS', 'CT') into separate binary columns. Ordinal Encoding is used when categorical values have a natural order, such as converting 'Diploma', 'Bachelor', and 'Master' into 0, 1, and 2 respectively.

As an example, a student with 'Attendance = 75%', 'Gender = Male', 'Course = CS', and 'Semester = 6' would be converted into numerical values that the Naive Bayes model could be processed. To improve the model's accuracy, preprocessing steps like handling missing values, label encoding, and one-hot encoding are applied to convert categorical data into numerical form. These preprocessing steps help reduce noise in the dataset and improve the overall performance and interpretability of the predictive model. The dataset before coding and the dataset after coding are shown below.

Table 2. Dataset Before Encoding

Gender	Marital status	Atte-mpt	Family Mem-ber	-	-	Dro p-out
Male	Married	1	3	-	-	No
Female	Single	2	5	-	-	Yes
Male	Single	0	5	-	-	No
Female	Single	1	6	-	-	No
Male	Single	1	5	-	-	Yes

Table 3. Encoded Dataset After Preprocessing

Gen -der	Marital status	Atte-mpt	Family Mem-ber	-	-	Drop - out
0	1	0	0	-	-	0
1	0	0	1	-	-	1
0	0	0	1	-	-	0
1	0	0	1	-	-	0
0	0	0	1	-	-	1

3.4. Decision Tree

A Decision Tree is a widely used machine learning algorithm for both classification and regression tasks. It works by recursively splitting the dataset into subsets based on the most significant feature at each level, forming a tree-like structure. Each internal node represents a decision based on a feature, each branch represents the outcome of the decision, and each leaf node represents a final class label or continuous value. Decision Trees are easy to interpret, handle both numerical and categorical data, and are effective in capturing non-linear relationships.

3.5. Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm commonly used for solving classification and regression tasks. It works by identifying an optimal decision boundary, known as a hyperplane, that best separates data points from different classes. The goal is to maximize the margin between the nearest data points (support vectors) of each class and the hyperplane, which increases the model's reliability when making predictions.

3.6. Naïve Bayes Classifier

Naïve Bayes is a classification method based on probabilistic reasoning, which applies 'Bayes' Theorem under the assumption that features are independent from one another given the class label. Despite this assumption often being unrealistic in real-world datasets, the algorithm remains highly effective due to its computational efficiency and ability to handle large datasets with many variables. The core idea is to estimate the likelihood that a data point belongs to a particular category by combining the prior probability of the class with the probability of observing each feature within that class. The formula can be expressed as:

$$P(Y\backslash Z) = \frac{P(Z\backslash Y).P(Y)}{P(Z)} \quad (1)$$

Here, $P(Y\backslash Z)$ represents the probability of class Y given the features Z, $P(Z\backslash Y)$ is the likelihood, $P(Y)$ is the prior probability of the class, and $P(Z)$ is the evidence term.

The following example describe the Naïve Bayes algorithm using a dataset of 24 records.
Prior Probabilities:
P (Dropout = No) = 16/24 = 0.67
P (Dropout = Yes) = 8/24 = 0.34

Likelihood for Dropout = No:
P (Dropout = No | Test Data)
=16/24 * 12/16 * 7/16 * 2/16 * 16/16 *14/16 * 11/16 * 14/16 * 9/16 * 4/16 *14/16 * 2/16 * 4/16 = 0.0000553
Likelihood for Dropout = Yes:
P (Dropout = Yes | Test Data) =8/24 * 6/8 * 2/8 * 2/8 * 5/8 * 1/8 * 1/8 *4/8 * 2/8* 1/8 * 7/8 * 1/8 * 4/8= 0.0000001304
Posterior Probabilities (Normalized):
P (Dropout = No | Test Data) ≈ 0.9976 (99.76%)
P (Dropout = Yes | Test Data) ≈ 0.00235 (0.235%)
Therefore, the final prediction is Dropout = "No".

Gender	Dropout	
	No	Yes
Male	7/16	2/8
Female	9/16	6/8

Attendance	Dropout	
	No	Yes
25-74	2/16	7/8
75-100	14/16	1/8

Marital status	Dropout	
	No	Yes
No	16/16	5/8
Yes	0/16	3/8

Mother Occupation	Dropout	
	No	Yes
No	14/16	7/8
Yes	2/16	1/8

Semester	Dropout	
	No	Yes
0-4	12/16	4/8
5-10	4/16	4/8

Age of Enrollment	Dropout	
	No	Yes
16-29	12/16	6/8
30-39	2/16	1/8
40-49	2/16	1/8

Father Occupation	Dropout	
	No	Yes
No	2/16	1/8
Yes	14/16	7/8

4. Results and Discussions

This system evaluated the effectiveness of three machine learning algorithms—Naive Bayes, Decision Tree, and Support Vector Machine (SVM)—in predicting student dropout. Using a dataset containing 500 student records with 13 features and 1 class of demographic and academic features, each model was trained using 80 % of data and tested using 20% of data. The key factors influencing dropout included attendance, academic performance, and socio-economic background.

The results revealed that both Naive Bayes and SVM achieved the highest accuracy of 86.0%, whereas the Decision Tree classifier yielded a slightly lower accuracy of 83.0%. This indicates that probabilistic models like Naive Bayes and margin-based classifiers like SVM are more suitable for handling complex, high-dimensional educational data compared to tree-based methods. Naive Bayes, in particular, was selected as the optimized model due to its simplicity, faster computation, and strong interpretability in educational settings.

Further analysis of classification outputs showed that the models were highly sensitive to attributes such as low attendance and poor grades. Students with below-average academic performance and low participation rates were consistently predicted as dropout-prone across all models. This confirms the importance of early identification of at-risk students through predictive analytics. Additionally, Naive Bayes provided an interpretable probabilistic framework that allowed for easy explanation of model decisions to educational stakeholders.

Overall, the system demonstrates the viability of using machine learning particularly Naive Bayes as a tool to support data-driven decision-making in education. These insights can be used by institutions to deploy targeted interventions and support services for students who exhibit risk factors. Future work may involve expanding the dataset, incorporating time-series performance data, or applying ensemble learning methods to improve predictive robustness.

Table 4. Accuracy Comparison Identification

Dataset	Feature	Method	Accuracy
Own Students Dropout Dataset (500)	14	Naïve Bayes	86.0%
		SVM	86.0%
		Decision Tree	83.0%

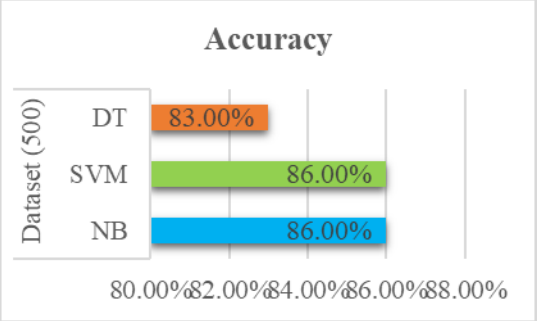


Figure 2. Comparison of Accuracy by different classifier

5. Conclusion and Future Works

In this study, a predictive model was developed using the Naive Bayes classifier to identify factors contributing to student dropout. The analysis was based on a dataset of 500 students comprising academic, demographic, and socio-economic attributes. The data used in this study, including student attendance, attempt, and semesters studied, were found to be significant indicators of student dropout risk. Through systematic preprocessing and model training, the Naive Bayes classifier achieved an accuracy of 86%, performing comparably with Support Vector Machine (SVM) and outperforming Decision Tree classifiers. The results highlighted attendance, academic performance, and family background as significant indicators of dropout risk. The strength of Naive Bayes lies in its simplicity, interpretability, and efficiency, making it an effective tool for educational institutions to identify at-risk students and intervene early.

To enhance the system's robustness and generalizability, several avenues for future research are proposed. Firstly, adding more data into the dataset from multiple academic years and institutions will improve the model's ability to capture broader patterns. Secondly, incorporating temporal data such as semester-wise performance or progression trends can enable time-series modeling for early dropout prediction. Additionally, exploring ensemble learning methods, such as Random Forest or Gradient Boosting, could further improve prediction accuracy. Finally, integrating the model into a real-time monitoring system with dashboards could support proactive interventions by educators and administrators.

References

[1] *M. Vaarma and H. Li, "Predicting student dropouts with machine learning: An empirical study in Finnish higher education,"* Technology in Society, *vol. 76, p. 102474, 2024.*

[2] *A. Meiriza, E. Lestari, P. Putra, A. Monaputri, and D. A. Lestari, "Prediction graduate student use Naive Bayes classifier," in* Proc. Sriwijaya Int. Conf. on Information Technology and Its Applications (SICONIAN 2019), Advances in Intelligent Systems Research, *vol. 172, pp. 370–375, 2020.*

[3] *Harwati, R. I. Virdyanawaty, and A. Mansur, "Drop out estimation students based on the study period: Comparison between Naïve Bayes and Support Vector Machines algorithm methods,"* IOP Conf. Ser.: Mater. Sci. Eng., *vol. 105, p. 012039, 2016.*

[4] *S. R. Sariman, H. Ab Jalil, and E. Marlisah, "Prediction model of school dropout factors using classification techniques in Selangor,"* Malaysian Journal of Social Sciences and Humanities (MJSSH), *vol. 9, no. 6, e002867, 2024.*

[5] *E. Yukselturk, S. Ozekes, and Y. K. Türel, "Predicting dropout student: An application of data mining methods in an online education program,"* European Journal of Open, Distance and e-Learning, *vol. 17, no. 1, 2014.*

[6] *L. T. Kunjumon, S. Shaji, S. T. Saji, T. Naushad, and N. Joseph, "An intelligent system to predict students' academic performance using data mining,"* Int. J. Inf. Syst. Comput. Sci., *vol. 8, no. 2, pp. 128–131, Mar.–Apr. 2019.*

Rice Classification System Using Deep Learning

Name Ma Soe Thinzar Mie
University of Computer Studies (Taungoo)
Taungoo, Myanmar
soethinzarmie97@gmail.com

Dr. Paing Thwe Soe
Department of Information Technology Supporting and Maintenance,
University of Computer Studies (Taungoo)
Taungoo, Myanmar
paingthwesoe@gmail.com

Abstract

Myanmar is a major rice-producing country with a diverse range of rice varieties. Since Burmese people mainly consume rice, it is necessary for the rice to be good. Rice image classification system developed with Myanmar rice varieties is currently not available in Myanmar's rice systems using the deep learning model. This paper represents to classify three different types of rice in Myanmar using Visual Geometry Group (VGG model). The system was tested on two different models (the original VGG16 model and the proposed VGG14 model) using a dataset of 30,000 images. The experiments on 30,000 images of Shwe Bo Paw San, Shwe Bo Kaut Nyin and Shwe Bo Aye Yar Min using modified VGG14 model with 30 epochs get 98.81% accuracy on training set and 98.80% accuracy in testing set. The experimental results show that the accuracy of the modified VGG14 model is better than the original VGG16 model.

Keywords: Deep Learning, Hybrid approach, Visual Geometry Group (VGG model), VGG16, VGG14.

1. Introduction

Rice is crucial for human economies, nutrition, and various industrial sectors. People in several areas utilize manual and physical procedures to classify rice grains [13]. This system makes a significant contribution to the field of rice classification using Deep Learning models [14].

DL models can identify abstract patterns in data through multiple layers of artificial neural networks, leading to more accurate and efficient results [5]. Furthermore, DL models' high adaptability allows them to learn and adjust to new and evolving datasets, making them suitable for diverse applications [7]. In DL, CNN stands for Convolutional Neural Network [6], which is a type of neural network specifically designed for images by extracting features using a process called convolution, making it particularly useful for image classification and object recognition [15].

This system used the Visual Geometry Group (VGG16) model [11]. Rice classification system was developed on deep learning modified VGG14 model to recognize and classify three different categories of Myanmar rice (Shwe Bo Paw San, Shwe Bo Kaut Nyin San and Shwe Bo Aye Yar Min.) Firstly, the images of three types of rice are collected by using an iPhone 12 Pro. It is collected using roughly 30000 photographs of three rice types, each of which has 10000 images in a dataset. Secondly, the VGG16 and VGG14 model were trained and tested. The experimental results show that the accuracy of the modified VGG14 model was better than the original VGG16 model.

2. Literature Review

The literature on rice classification using deep learning and machine learning was reviewed in this part, along with another essential research.

Farshad Farahnakian *, Javad Sheikh, Fahimeh Farahnakian, Jukka Heikkonen (2023) investigated the effect of CNN Deep learning on the performance of deep-learning models, including Residual Network (ResNet), Visual Geometry Group (VGG) network, EfficientNet, and MobileNet. The authors used the rice image dataset, which consists of 75,000 images, classified into five different rice categories. The authors showed that the EfficientNet-based model delivered the highest accuracy (99.67%), 99.53% in VGG16, 99.57% in VGG19, ResNet50 99.53%

accuracy, and MobileNet 98.86%, respectively [1]. This paper only uses an existing model on an existing dataset and does not create a new model.

In 2016, authors from Myanmar utilized image processing and machine learning to classify five varieties of Myanmar rice such as Paw San Hmwe, Lone Thwe Hmwe, Ayeyarmin, Kauk-Nyinn-Thwe and Kauk-Nyinn-Pu [2]. This system is effective in Myanmar because Myanmar is great producer of different qualities of rice grains and therefore the study and basic implementation would greatly help the researchers, agriculturist and other stakeholders of agricultural growth. According to the authors, this paper only involves manual classification and is not very efficient. The reason is that only 38 images were used in the study.

In Vietnam, authors Nguyen Hong Son and Nguyen Thai-Nghe used a deep learning algorithm on 2,000 images to classify the quality of one type of Vietnamese rice. Experimental results on 2000 real images of whole rice and broken rice using CNN get 99.16% accuracy on training and 89.75% accuracy in testing [3]. The system also shows the results on testing set of 85.06% accuracy in SVM-HOG and 84.30% accuracy in KNN, respectively. In this paper, only one type of Vietnamese rice was used, and the classification was limited to distinguishing between whole rice and broken rice.

Thae Nu Wah, Pann Ei San, and Thandar Hlaing Technological University (Thanlyin) Republic of the Union of Myanmar (2018) proposed classification and analysis on feature extraction of rice grain for Myanmar rice. In this study, three classes of Paw San Mhwe rice grain in Myanmar were considered for classifying and grading. Specified features of the objects are extracted and classified the rice classes by using KNN (K-nearest neighbor) classifier. When testing Paw-San rice into three classes, the accuracy for Class A is (100%), for Class B is (93%) and for Class C is (83%) [4]. Only 90 images were used, which is not sufficient to thoroughly test the classification and grading of rice quality.

3. Methodology

This system includes two parts of experimental implementation using two datasets. In the first part, the original VGG16 model was used to train on own dataset containing 30,000 images, and the results were presented. In the second part, a modified VGG model with 14 layers was proposed, and it was used to perform training and testing on own dataset. Finally, a comparison of the results between VGG16 and VGG14 was conducted.

3.1. Dataset Collection

This system utilized two datasets: an existing Kaggle dataset and a custom-built dataset. The Kaggle dataset includes 75,000 images across five rice classes [10]. The custom dataset comprises 30,000 RGB images of three Myanmar rice varieties: Shwe Bo Paw San, Shwe Bo Kaut Nyin San, and Shwe Bo Aye Yar Min, captured using an iPhone 12 Pro. Each rice type includes 10,000 single-grain images. All images in the custom dataset were preprocessed by resizing them to a fixed resolution of 150×150 pixels to ensure uniform input dimensions for the model. The dataset was split into 80% for training (24,000 images) and 20% for testing (6,000 images).

3.1.1. Data Augmentation

To enhance the generalization capability of the model and reduce overfitting, data augmentation techniques were applied to the training dataset. These techniques artificially increase the size and diversity of the dataset by applying random transformations to the input images. The following augmentation parameters were used: a rotation range of 20 degrees, width and height shift ranges of 0.2, shear range of 0.2, and zoom range of 0.2. Additionally, horizontal flipping was enabled. These transformations help the model become invariant to minor distortions and improve its performance on unseen data.

3.2. VGG16 Model

VGG-16 is a convolutional neural network that is 16 layers deep [8]. Instead, the model was trained from scratch using the custom Myanmar rice dataset, which includes 30,000 RGB images. VGG16 uses convolution layers with a 3x3 filter and a stride 1 that are in the same padding and maxpooling layer of 2x2 filter of stride 2 [12]. In the end it has two fully connected layers, followed by a softmax for output. This is shown in Figure 1.

In Convolutional Neural Networks (CNNs) [9], the convolution layer serves as the core component responsible for extracting features from input images using filters or kernels. To introduce non-linearity, the ReLU (Rectified Linear Unit) activation function is applied, enhancing the network's learning capability. The pooling layer, particularly max pooling, reduces spatial dimensions and computational cost while preserving important features. A flatten layer then converts feature maps into a one-dimensional vector, connecting to the fully connected (FC) layer, which performs the final classification. To prevent overfitting and improve generalization, a dropout layer is used during training.

To enhance the training dataset and improve model generalization, the ImageDataGenerator class from Keras was utilized with a comprehensive data augmentation strategy. The generator was configured to normalize input images by rescaling pixel values to the [0, 1] range using rescale=1/255. A validation split of 20% was defined to allocate a portion of the data for validation purposes. Various augmentation techniques were applied, including random rotations up to 10 degrees (rotation_range=10), horizontalshifts(width_shift_range=0.1),verticals hifts(height_shift_range=0.1), shearing transformations (shear_range=0.1), and zooming (zoom_range=0.1). Additionally, horizontal flipping (horizontal_flip=True) was enabled to introduce mirror-image variations. These augmentations effectively reduce overfitting by generating diverse and realistic variations of the training data.

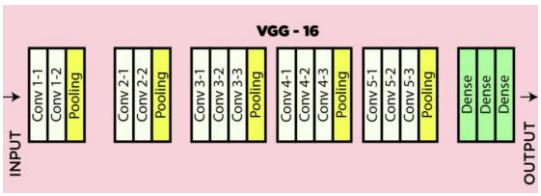


Figure 1. VGG16 architecture map

3.3. Proposed VGG14 Model

The proposed VGG14 model is a custom modification, inspired by the original VGG16 architecture but not an officially recognized VGG variant. This custom VGG14 architecture was designed to optimize performance on rice classification by reducing the depth of the network and adjusting layer configuration for better domain-specific learning.

In Figure 2, VGG14 architecture has thirteen convolution blocks, and one dense layer. Key modifications include reducing the total number of layers from 16 to 14, using only one fully connected layer instead of two, and introducing a dropout layer before the FC layer to improve generalization. This architecture was chosen to achieve lighter computation and faster convergence without sacrificing classification performance. Experimental results confirm that the custom VGG14 model outperformed VGG16 on both training and validation sets.

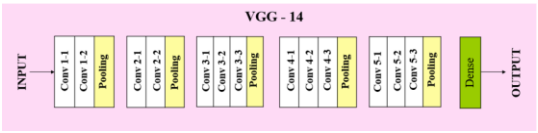


Figure 2. VGG14 architecture map

3.4. Model Training

In Figure 3, the proposed system includes seven steps. Initially, image datasets, both a standard Kaggle dataset and a custom-built dataset are prepared. In the next step, image resizing and augmentation are applied to ensure consistency and enhance model performance. The VGG16 and VGG14 architectures are then constructed and trained using the prepared datasets.

After training and validation, both models are compiled and fitted. Their performance is analyzed through accuracy and loss metrics visualized with line graphs. Finally, evaluation metrics are used to assess each model, and the one with the highest accuracy is saved as the final trained model.

3.5. Model Testing

When a user enters a new rice image into this system, preprocessing is made by resizing the image (150 x 150 pixels). And the trained model (Hierarchical Data Format(.h5) Model) was loaded. Then, this image was classified into predefined classes such as Shwe Bo Paw San Mhwe, Shwe Bo Aye Yar Min or Shwe Bo Kaut Nyin San. Finally, the predicted result and image were displayed. This is shown in Figure 4.

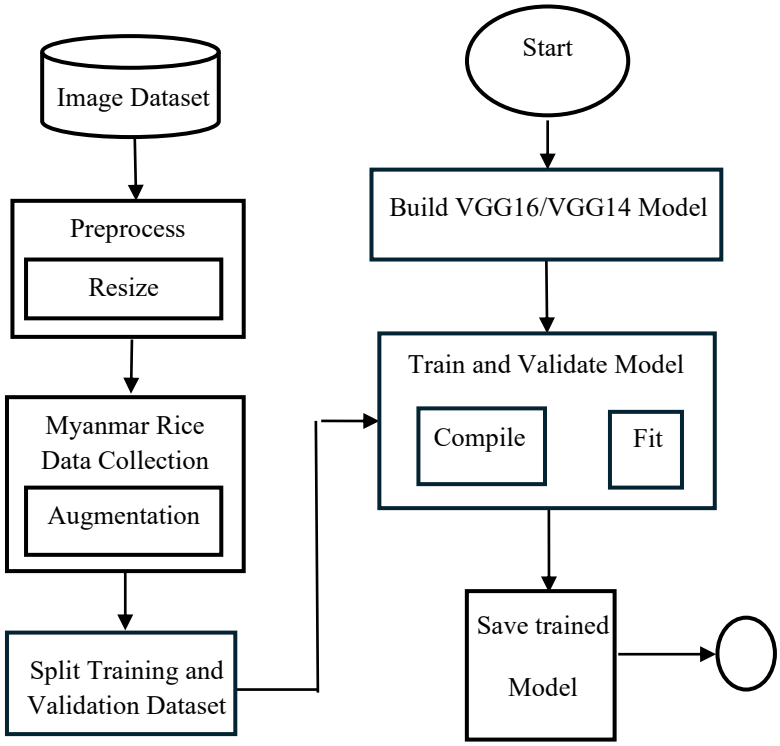


Figure 3. Flowchart of the proposed train and validate model

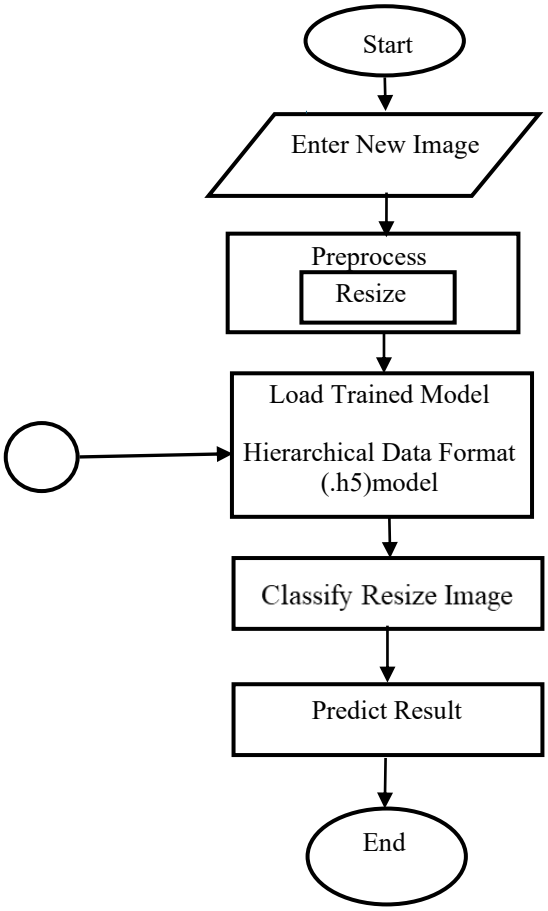


Figure 4. Flowchart of the proposed testing model

4. Experimental Results

Classification results using VGG14 and VGG16 models are given in this section. The dataset used in the study contains features obtained from 30,000 rice grain images. In VGG14 and VGG16 models, these data were used as input. Shwe Bo Paw San, Shwe Bo Kaut Nyin San and Shwe Bo Aye Yar Min rice classes were given as classification outputs. The experiments on 30000 images using VGG14 model get 98.81% accuracy on training set and 98.80% accuracy in validation set. VGG16 model was also used to classify and the accuracy results of 33.36% and 33.33% were achieved on testing set. The experimental results are presented in Table 1.

Since the VGG14 model has two fewer layers, there was a concern that it might only perform well on the custom dataset. To address this, further testing was conducted on the standard dataset (Kaggle Dataset**)** to verify that its performance across different datasets. The analyses on standard datasets using VGG14 model get 98.89% accuracy on the training set and 99.30% accuracy in testing set. The most effective classifier was based on the efficiency and accuracy of all two techniques. VGG14 (98.89%), VGG16 (20.09%) accuracy. VGG14 model obtained the highest accuracy of 98.89% among the classifiers. The results confirmed that the model still achieves good accuracy on the standard dataset, indicating its robustness. This is shown in Table 2.

Since this system emphasized on Myanmar rice dataset, the custom dataset was focused. The VGG14 model attained the classification success rate of 98.81%. The analysis results revealed that VGG14 model gave better accuracy than VGG16 model. Based on the results, VGG14 model provided better results are shown Table 1 and Table 2. So, the VGG16 model with 14 layers outperformed achieving better accuracy and lower loss.

VGG14, having fewer layers, may have been more appropriate for the resolution (150x150) and the relatively low complexity of rice grain features. During training, VGG14 converged faster and reached high accuracy with lower validation loss, indicating stable learning. Thus, the VGG14 model showed better accuracy, generalization, and computational efficiency in this application.

Table 1. Comparing the accuracy and loss results of two different models VGG16 and VGG14 on Own Dataset

No	Layers	Epochs	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Run Time	Date
1	14 layers	30 epochs	0.9881	0.9880	0.0352	0.0356	8 hours	19.8.2024
2	16 layers	20 epochs	0.3366	0.3333	10.6927	10.7454	6:30 hours	23.8.2024

Table 2. Comparing the accuracy and loss results of two different models VGG16 and VGG14 on Kaggle Dataset

No	Layers	Epochs	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Run Time	Date
1	14 layers	10 epochs	0.9889	0.9930	0.0347	0.0203	7 hours	26.11.2024
2	16 layers	10 epochs	0.2009	0.2000	9.6785	9.6709	8 hours	27.11.2024

5. Limitation

The system was limited because three different kinds of rice were only used. And the model's applicability was restricted to a wider variety of rice varieties. The dataset included photographs with a range of resolutions, some of which are less than ideal, because high-quality images were always unavailable and lower-

resolution photos could not include the fine details required for precise classification, this could potentially have an impact on the model's performance.

6. Conclusion and Future Work

The proposed VGG14 model was used to recognize and classify three difference varieties of rice (Shwe Bo Paw San, Shwe Bo Aye Yar Min and Shwe Bo Kaut Nyin San). The custom dataset includes three types of rice grains, with each grain containing 10,000 images. The experimental results for 30000 images of rice grain illustrated that the proposed approach works well in accuracy. These architectures can be successfully applied to solve the challenges in the rice grain images. This paper reviewed CNN-based techniques for classifying Myanmar rice images. Though many architectures have been proposed to solve rice image classification, the current paper only focused on deep learning-based methods. The VGG14 model can be applied to classify Myanmar rice images. The current system was only deployed with one deep learning model and was based on 30,000 images. In the future, this system can be expanded to other rice varieties and utilized a large amount of training and testing dataset.

References

[1] M. Manataki, N. Papadopoulos, and E.F. Roberts, “A Comparative Study of AlexNet and VGG on a Dataset from Archaeological Sites”, June 2023, n.p.

[2] M.M. Tin, K.L. Mon, E.P. Win, and S.S. Hlaing, *Myanmar Rice Grain Classification Using Image Processing*, Myanmar, 2016, pp. 1-5.

[3] N.H. Son, and N.T. Nghe, *Deep Learning for Rice Quality Classification*, ACOMP, Vietnam, November 2019, pp. 92-93.

[4] T.N. Wah, P.E. San, and H. Hlaing, “Analysis on Feature Extraction and Classification of Rice Kernels for Myanmar Rice Using Image Processing Techniques”, Myanmar, August 2018, pp. 603-604.

[5] F. Farahnakian, J. Sheikh, F. Farahnakian, and J. Heikkonen, “A comparative study of state-of-the-art deep learning architectures for rice grain classification”, Journal of Agriculture and Food Research, Finland, March 2024, vol. 15, n.p.

[6] R. Yamashita, M. Nishio, R.K.G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology”, 22 June 2018, pp. 1-21.

[7] J. Rane, S.K. Mallick, and N. Rane, *Scalable and adaptive deep learning algorithms for large-scale machine learning systems*, Deep Science Publishing, October 2024, pp. 40-41.

[8] R. G, Everything you need to know about VGG16, Sep 23,2021, n.p.

[9] A. S. Gillis, What is a convolutional neural network (CNN), Nov25 2024, n.p.

[10] M. Koklu, Rice Image Dataset, 2021, n.p.

[11] VGG-16 | CNN model, 21 Mar 2024, n.p.

[12] M. Tripathi, Analysis of Convolutional Neural Network based Image Classification Techniques, Journal of Innovative Image Processing, June 2021, vol. 3, pp. 100-108.

[13] B. Arora, S. LR, S. Arcot, Rice Grain Classification using Image Processing & Machine Learning Techniques, 2020 International Conference on Inventive Computation Technologies (ICICT), February 2020, n.p.

[14] N. Prakash, R. Rajakumar, N. Leela Madhuri, M. Jyothi, A. Pavithra Bai, M. Manjunath, K. Gowthami, Image Classification for Rice varieties using Deep Learning Models, vol. 21, June 2022, pp. 261-263.

[15] Z. Kelta, An Introduction to Convolutional Neural Networks (CNNs), Nov 14, 2023, n.p.

Violence Detection System Using MobileNetV2 and Bidirectional-GRU

Thet Hsu Myat, Darli Myint Aung
University of Computer Studies (Taunggyi)
thethsumyat@ucstgi.edu.mm , darlimyintaung@ucstgi.edu.mm

Abstract

Nowadays, safety and security are critical for maintaining a safe environment and preventing criminal cases. There is need to be secure in various environments (campus, airport, mall, etc.). Safety and security system are crucial for individuals and everywhere. Traditional surveillance systems depending on manual monitoring are often time-consuming to watch the tremendous number of videos. To address this, this paper proposes an automatic violence detection system using MobileNetV2 for spatial feature extraction and Bi-GRU for temporal feature extraction. The proposed system preprocesses video data by extracting and normalizing frames, then trains a model to classify violence or non-violence activities. This system obtains training accuracy 88.12% and validation accuracy 85.03% on this dataset.

1. Introduction

In recent years, the video surveillance system, also known as CCTV, is widely used in airports, train stations, shopping malls, universities and several places. This system is utilized to monitor violence or abnormal activities, secure safety and prevent criminal cases in these areas. This camera captures footage which can be viewed live or stored for replay later. However, this camera produces a huge amount of video that it is difficult to manually find violent activities from the tremendous amount of video.

Traditional surveillance systems usually concern manual monitoring through CCTV cameras by security teams. As the increased number of cameras becomes difficult to monitor every part all the time. These systems have limitations as they require constant human monitoring, and it can be tiresome to detect.

With the rise of technology, video violence detection systems have provided more effective than traditional security measures. Video violence detection system is a task of recognizing and detecting that deviate from normal behaviors. This detection system has crucial importance in healthcare, industrial, smart homes, university campuses, public safety (e.g. traffic monitoring and transportation), financial, military and data centers, etc. The systems can detect violence activities that may compromise security teams. The system is more efficient than people watching CCTV all the time. Therefore, there is an intelligent system that can assist security teams.

In this paper, the violence detection system using MobileNetV2 in Convolutional (CNN) and Bi-Gated Recurrent Unit (GRU) in Recurrent Neural Network (RNN). The aim of the proposed system is to help people do things more easily as well as reduce the reliance on human monitoring.

This system is presented in the remainder of this paper. The paper is organized as follows: Section 1 gives introduction, section 2 describes different related work, section 3 presents proposed system. In section 4, the implementation and experimental results. Finally, Conclusion are summarized in section 5.

2. Related Works

Various methods have been used to detect different types of anomalies. However, each method has been developed for use in a specific environment. The following sections discussed some of the previously used models in detail, emphasizing their challenges and the accuracy attained.

Their system created the dataset with normal and abnormal events captured by CCTV footage that is 128 hours in length. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are used to detect suspicious activities in real-time. The Inception V3 model in CNN is a pre-trained model and high-level features from images, which are then processed by the RNN to

recognize temporal features and anomalies. The system classified two categories such as Threat (abnormal activities) and Safe (normal activities). Their system achieved 97.23% accuracy and reduced overfitting [6].

The system uses ResNet architecture to extract spatial features from video frames and SRUs to extract temporal features. Three alternatives of the model are explored in this system: ResNet18 + SRU, ResNet34 + SRU, and ResNet50 + SRU. The system is evaluated using the UCF-Crime dataset with 88.92%, 89.34% and 91.24 % accuracy respectively. The results showed that ResNet-50 with SRU is the most accurate than the other two [4].

Their system used PCANet and CNN, addressing labor-intensive processes and missing abnormal behaviors in crowded environments. Their system used UMN dataset, Avenue dataset, and UCSD dataset for anomaly detection and the UCF 101-Action Recognition dataset for action recognition. Principal Component Analysis (PCA) is used for feature extraction from video frames, while Support Vector Machine (SVM) classifies frames as normal or abnormal based on these features [2].

Long Short-Term Memory (LSTM) autoencoders model is used for real time surveillance systems on campuses. The system contained two stages, the features extracted from surveillance footage in the first stage, then classified the abnormal behavior in the second stage and responded to alert via an ESP32 microcontroller for notifications. The model achieved an anomaly detection loss of 4.795 on the UCSD anomaly dataset [5].

The comprising of deep Multiple Instance Learning (MIL) and Inflated 3D Convolution Network (I3D-ResNet50) is used on UCF-Crime dataset. I3D-ResNet50 is a pre-trained model that is used for feature extraction with 10-crop augmentation. The performance of the proposed model increases the variety of trained data, the image augmentation methods, dropout regularization with 30% between Fully Connected Neural Network (FCNN) are used. Their experimental results attained the best accuracy that display 82.85% AUC score compared with another methods [1].

Their system used pre-trained 3DConvNet for feature extraction and hand-craft datasets with training data of 800 normal and 810 abnormal videos and the testing data of 150 normal and 140 abnormal videos. In that paper, videos are resized to 240*320 pixels and extract 30 frames per second. The experimental results show that the proposed model operates better than basic methods [7].

3D-CNN and Optical Flow is utilized on RWF-2000 dataset. Their proposed methods with four stages such as RGB channel for extracting spatial features, Optical Flow channel for capturing motion dynamics, Merging Block for fusing information from RGB and optical flow, and Fully Connected Layers for classification the results. The system achieved 87.25% accuracy [3].

3. Proposed System

In this section, this system is proposed the violence detection system using Mobile Net V2 with Bi-Gated Recurrent Unit (GRU) to detect violence behavior in video.

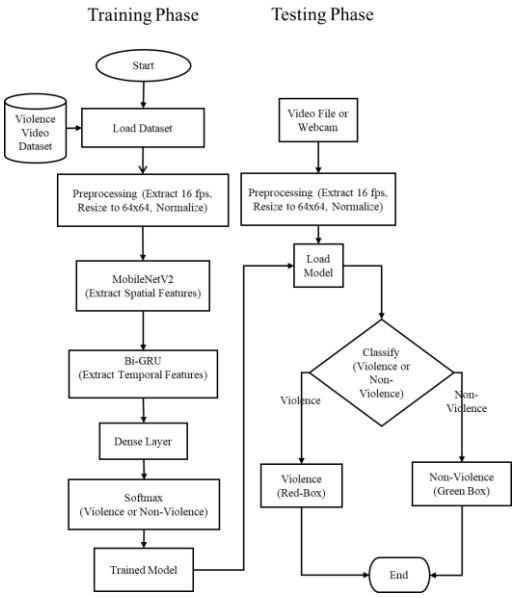


Figure 1. Proposed System Flowchart

Figure 1 shows the workflow of the proposed violence detection system. In the training phase, it starts by loading a dataset and then the video is preprocessed by extracting frames, resizing and normalizing them. MobileNetV2 extracts spatial features and Bi-GRU layer extracts temporal features between frames. These frames go through dense and softmax layers to classify violence or non-violence. In the testing phase, the model can detect new input video or webcam and then classify a red box for violence or a green box for non-violence after the model is trained.

3.1. Preprocessing

In pre-processing phase, each video is passed through a frame extraction process using OpenCV, where 16 frames per second from the video. These frames are then resized to the same dimensions of 64x64 pixels and normalize their pixel values between 0 and 1.

The sequence of pre-processed frames is stored and labelled after preprocessing. All sequences and labels are inserted into numpy array (features numpy and labels numpy) which are divided into two phases (training and testing). It split function as 80:20. These labels are converted into one-hot encodings for the task of classification.

3.2. Training Phase

In this phase, the model uses a Time Distributed wrapper around a pretrained MobileNetV2 model to extract spatial features from every frame. The MobileNetV2 model, which is pretrained on ImageNet, is polished by gradually unfreezing the last 40 layers for better violence detection task performance.

A pre-trained MobileNetV2 extracts spatial features from individual frames and the extracted features from each frame sequence are passed into a Bidirectional GRU (Bi-GRU) network with 64 units, which captures temporal dependencies across the sequence in both forward and backward directions. The GRU output is then passed through fully connected layers with Dropout regularization to reduce overfitting. The first Dense layer has 256 units with ReLU activation and 0.3 dropout rate and then this passed through with 128-unit Dense Layer, 64-unit Dense layer by 0.3 dropout rate. Finally, the output layer uses SoftMax activation to classify the video as either Violence or Non-Violence.

3.3. Testing Phase

In the testing phase, the proposed violence detection system is built using GUI (in figure 2) with Tkinter Library where the trained model is tested via live webcam input or video files for detecting violence or non-violence. The users can detect by selecting a video or live webcam. When video or webcam is chosen, the system reads each frame, resizes and stores 16 frames per second. Each frame passed through the trained model for prediction. And then, the model determined "violence" or "non-violence" on the frames.

Figure 2. Violence Detection GUI

4. Methodology

4.1. MobileNetV2 Architecture

The MobileNetV2 architecture is based on bottleneck depth separable convolutions and is constructed using residual connections composed of basic blocks. Mo-bileNetV2 includes two main types of layers as shown in figure 3. The first layer is "ReLU6" along with 1*1 convolutions, and the second layer is Depth wise convolutions. The third layer also uses 1*1 convolutions, but it does not use a non-linear activation function.

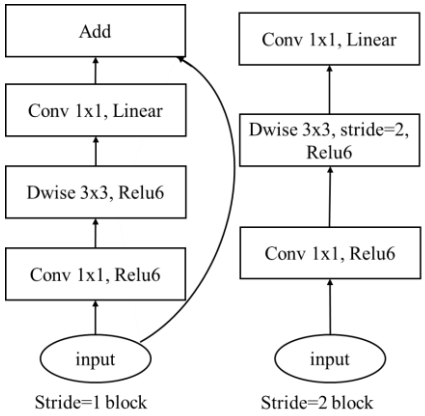


Figure 3. MobileNetV2 Architecture

The first layer is only used stride 1 and the second only use stride 2 to shrink. The method of object detection is to do categorize, to define input class, regression to modify bounding box. The omission of the final connected layers is complete, detection of backbone networks for classify works. The network of backbone serves as extracting feature of object detection works, capturing images for input and manufacturing feature maps for every single input image.

The pretrained model is used to enable effective feature extraction, particularly for classification tasks. MobileNetV2 often uses "ImageNet" pretrained weights. ImageNet is a

large-scale dataset containing millions of labelled images and it makes effective use of image classification. It is a database of an image which has been trained upon thousands of images. And it is useful for categorization of images.

Table 1. Residual block in k to k' channel

Input	Operator	Output
h × w × k	1x1 conv2d, ReLU6	h × w × (tk)
h × w × k	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	Linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 1 represented input, operand and output with residual block in k to k' channel [9].

4.2. Bidirectional GRU

A Bidirectional Gated Recurrent Unit (Bidirectional GRU or Bi-GRU) is an extension of the standard GRU. Bi-GRU consists of two GRU Layers: One GRU works the sequence forward and another GRU works the sequence backward directions. The input and forget gates are the only features of bidirectional recurrent neural networks. Bidirectional GRU architecture consists of Input Layer, Forward GRU Layer, Backward GRU Layer, Concatenation Layer and Output Layer as shown in figure 4.

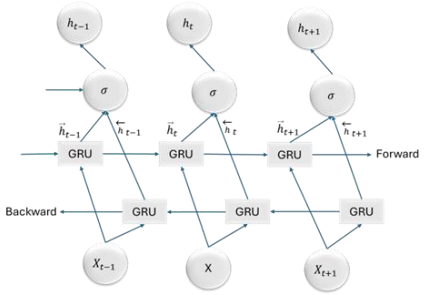


Figure 4. The Architecture of Bidirectional GRU

For a given Input Sequence $x = (x_1, x_2, \ldots, x_T)$

- Forward GRU computes hidden states:

$$h_t^{\rightarrow} = GRU\ (x_1, h_{t-1}^{\rightarrow}) \quad (1)$$

- Backward GRU compute:

$$h_t^{\leftarrow} = GRU\ (x_1, h_{t+1}^{\leftarrow}) \quad (2)$$

- Final Output at time step t is:

$$h_t = [h_t^{\rightarrow}, h_t^{\leftarrow}] \quad (3)$$

- GRU involves:

Updated gate:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (4)$$

Reset gate:

$$r_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (5)$$

Candidate hidden state:

$$\tilde{h}_t = \tanh\ (W_h x_t + U_z(r_t \odot h_{t-1}) + b_h) \quad (6)$$

Final hidden state:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (7)$$

Where, σ= the sigmoid function, ⊙=element-wise multiplication, W, U, b are learnable parameters [12].

5. Implementation and Experimental Results

5.1. Dataset

In this paper, the system is implemented on Real-Life Violence Dataset, GitHub datasets and the created datasets. There are a total number of 2628 videos which are collected from Kaggle, GitHub and a custom-collected campus dataset. The created dataset consists of a collection of videos that record various students' activities where students are fighting each other and walking together with friends or other normal social interactions. The dataset consists of 1310 non-violent videos and 1318 violent videos. The dataset is divided into three parts: testing, training and validation. From the datasets, these data were split into training and testing (80% and 20%) respectively. The aim of split data is to avoid overfitting. This includes 1048 non-violence and 1054 violence videos for training, and 262 non-violence and 264 violence videos for validations. This system emphasizes violence (fighting) and non-violence behaviour. Therefore, the combining datasets is more accurate results and accuracy.

The sample of violence and non-violence behaviours are shown in figure 5 and 6.

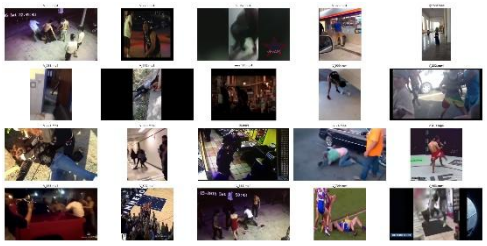


Figure 5. Sample dataset of Violence Behavior

Figure 6. Sample dataset of Non-Violence Behavior

5.2. Results Discussion

The model is built using MobileNetV2 with Bi-GRU for detection process. The proposed system achieved training accuracy 88.12% and validation accuracy 85.03% with 0.0036 learning rate. The proposed model is considered by evaluating the precision, recall and F1-scores in Figure7.

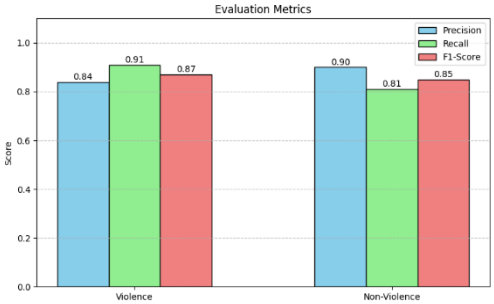
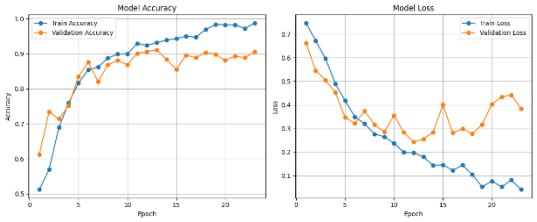


Figure 7. Evaluation Metrics



(a) (b)

Figure 8. Model Loss and Accuracy

Figure 8 displays the model accuracy and loss throughout each epoch. In (a), the accuracy increased significantly from epoch 1 to 14 and the model improvements slow after epoch 14. In (b), training and validation loss going down from epoch 1 to 10. Starting from epoch 10, the validation loss starts to fluctuate.

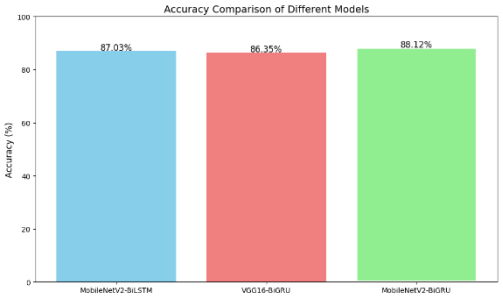


Figure 9. Performance evaluation of different models: MobileNetV2-BiLSTM, VGG16-BiGRU, and Proposed MobileNetV2-BiGRU

The accuracy of three models: a baseline MobileNetV2-BiLSTM, VGG16-BiGRU and the proposed MobileNetV2-BiGRU are compared as shown in figure 9. According to the result, the proposed model achieved 88.12%. Meanwhile, the MobileNetV2-BiLSTM and VGG16-BiGRU have 1.09% and 1.77 % gap with the proposed model. The proposed model has better accuracy than the other two models.

5.3. Comparison Results of Various Distance from Webcam

Figure 10. Violence Detection by Displaying a Video to the Webcam

The system can detect and classify real-time activities via webcam as shown in figure (10,11,12). It can identify non-violence or violent behaviour by testing live footage frames. The system displays non-violence activities with a green box and the violence activities with a red box.

Therefore, this system quickly detects harmful and dangerous situations. By using webcam, the detection system is simple, real-time and practical for security and monitoring processes.

In figure (10), the system detects violence or non-violence by playing a video in front of the webcam. In figure (11) demonstrates the violence detection using webcam at 3 feet distance. In figure (a), two girls are talking normally which is non-violence behaviour and is showed with a

green box. Figure (b) shows the girls are pushing each other that indicated the start of aggressive behaviour, and figure (c) shows one girl is hitting another that clearly showed violence behaviour. Both (b) and (c) show violence behaviour, the system detects them with a red box. These results highlight the system's ability to classify behaviours at 3 feet range from webcam.

(a) (b) (c)

Figure 11. Violence Detection via Webcam at 3 feet

(a) (b) (c)

Figure 12. Violence Detection via Webcam at 6 feet

Figure 12 demonstrates the webcam's detection of behaviours at 6-feet range. (a) shows a normal interaction. In (b), they are fighting each other, but the system has difficulty detecting the violence clearly because they are farther away from webcam. But people are trying to hit each other in (c), the system detects the violence behaviour. This highlights potential challenges in accurate detection at 6 feet or greater distances.

Table 2. Limitations

Distance from webcam	Detection Performance
3-feet	Well detected
6-feet	Misses or delay detection

According to table 2, although the system can detect violence or non-violence around 3 feet, it can miss, or delay detection exceed to 6 feet because of the web cam limitation.

6. Conclusion

Violence Detection System is important to monitor and prevent criminal behaviors in various situations, to ensure public security. In this paper, the violence detection system using MobileNetV2 with Bi-GRU obtained training accuracy 88.12% and validation accuracy 85.03% on comprising dataset. MobileNetV2 extracted spatial features and Bi-GRU extracted temporal features. The system can work both video files and webcam, but it may face some limitations. In the future, we will improve robustness under different lighting and occlusion, and enhance by automatically detection such as abuse, fighting, robbery and vandalism, etc. via CCTV.

References

[1] A. Elmetwally, Reem Eldeeb, and Samir Elmougy, "Deep learning-based anomaly detection in real-time video," *Multimedia Tools and Applications*, May 2024, doi: https://doi.org/10.1007/s11042-024-19116-9.

[2] A. Mohan, M. Choksi, and M. A. Zaveri, "Anomaly and Activity Recognition Using Machine Learning Approach for Video Based Surveillance," Jul. 2019, doi: https://doi.org/10.1109/icccnt45670.2019.8944396.

[3] M. Cheng, K. Cai, and M. Li, "RWF-2000: An Open Large Scale Video Database for Violence Detection," *arXiv.org*, 2019. https://arxiv.org/abs/1911.05913 (accessed May 28, 2025).

[4] M. Qasim and E. Verdu, "Video anomaly detection system using deep convolutional and recurrent models," Results in Engineering, p. 101026, Mar. 2023, doi: https://doi.org/10.1016/j.rineng.2023.101026.

[5] M. Sandler, A. W. Howard, M. Zhu, Andrey Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv (Cornell University)*, Jan. 2018, doi: https://doi.org/10.48550/arxiv.1801.04381.

[6] N. Tuteja, R. Sharma, and C. Nagar, "Real-Time Anomaly Detection Surveillance System," © *2023 IJNRD* |, vol. 8, no. 6, p. 566, 2023, Available: https://www.ijnrd.org/papers/IJNRD2306358.pdf

[7] Tiya Vaj, "GRU vs. Bi-GRU: which one is going to win? - Tiya Vaj - Medium," *Medium*, Feb. 02, 2023. https://vtiya.medium.com/gru-vs-bi-gru-which-one-is-going-to-win-58a45ede5fba

[8] V. Singh, S. Singh, and P. Gupta, "Real-Time Anomaly Recognition Through CCTV Using Neural Networks," Procedia Computer Science, vol. 173, pp. 254–263, 2020, doi: https://doi.org/10.1016/j.procs.2020.06.030.

[9] W. Sultani, C. Chen, and M. Shah, "Real-world Anomaly Detection in Surveillance Videos." Available: https://openaccess.thecvf.com/content_cvpr_2018/papers/Sultani_Real-World_Anomaly_Detection_CVPR_2018_paper.pdf

Travel Package Purchase Prediction System Using Ensemble Learning Algorithms

Thae Su Hlaing, Nan Saw Kalayar
University of Computer Studies (Taunggyi)
thaesuhlaing@ucstgi.edu.mm, sawkalayar@ucstgi.edu.mm

Abstract

Myanmar has many famous pagodas, places and historical sites for tourism. It must be developed sustainably to maximize the advantages for the country from tourism-related income such as job opportunities, foreign currency earnings while maintaining its valuable cultural and natural heritage because tourism plays an important role in the local economies of many nations, including Southern Shan State. The tourism industry is now using data and technology to offer a better service after understanding the customer behaviors. However, there is a limited attention to predict the travel package purchase behavior especially in Southern Shan State region. The proposed machine learning-based system to predict a customer will purchase a travel package or not among the predefined five packages by analyzing the socio-demographic and travelling behaviors. In this system, the data are collected through closed-encoded survey questionnaires and then preprocessing the data to speed up the system accuracy. There are total 12 features are used in this system. Random Forest and Extreme Gradient Boosting (XGBoost) are trained and their performance are measured using accuracy, precision, recall, F1-score, and ROC-AUC. The proposed XGBoost model and Random Forest achieved the accuracy of 97% using the dataset of 500 samples. And the accuracy of proposed system is compared with other classification models such as Logistic Regression and Decision Tree. Moreover, the accuracy of model is described after using SMOTE data balancing and Hyperparameter Tuning techniques.

1. Introduction

Tourism plays an essential role in the socio-economic development of regions like Southern Shan State, Myanmar, which is famous for its scenic beauty, cultural heritage, and local traditions. However, the tourism industry often struggles with forecasting customer behavior and personalizing offers. Predictive systems using machine learning offer a powerful tool to address this gap by analyzing customer data and identifying patterns that influence travel package purchases.

The aim of this proposed system is to design and develop a travel package purchase prediction system that can accurately forecast whether a potential customer is likely to purchase a travel package. The system leverages machine learning algorithms: Random Forests and XGBoost to predict customer decisions based on various features such as customer demographics, travel preferences, and economic status.

Most previous research in the tourism sector has used general-purpose datasets (like from online platforms or travel websites) and focused on recommending places system based on where people have gone before. This system focuses on a different but important question "Will a customer actually purchase a travel package or not?" This system uses a combination of six socio-demographic features (such as age and income) and four travel behavioral features (such as number of trips in a year) to provide a predictive model using Random Forest and XGBoost Algorithms.

This work addresses contributes to the growing field of travel analytics, which refers to the application of data analysis and machine learning techniques to understand and predict patterns in the tourism and travel industry. Travel analytics is commonly used to enhance customer experiences, to improve service planning and support marketing decisions in tourism sectors.

This system focuses on Southern Shan State, a culturally and environmentally rich region known for destinations like Inle Lake, Kalaw, Ywangn and Pindaya. By collecting and analyzing data specific to this local context, this system introduces original, localized insights that have not been thoroughly studied before.

This proposed system could be helped tour operators to identify and target on the right customers, and improve their marketing strategy.

2. Related Works

J.C.Sancho Núñez presented a comprehensive analysis of how machine learning techniques are being increasingly applied within the tourism sector to address challenges in areas such as planning, forecasting, recommendation, prevention, and security. The author provided insights into the most effective ML methods and their suitability for different tourism-related problems [4].

I.Pinto studied about the tourist purchasing behavior through online travel agencies (OTAs) like Booking.com searched the main factors that influence tourists' purchase decisions when booking accommodations online. The price is the most critical determinant in the decision-making process in this study. Moreover, this study shown that a traveler's socio-demographic data like age, income, and where they live can influenced the buying decision process [3].

J.L.Nicolau and F.J.Mas presented a model that show the process of tourist choice in two stages_(i) the decision to go on a holiday, (ii) the level of holiday expenditures by using the Heckit model. The author analyzed a sample of 3,781 individuals in Spain and demonstrate that these decisions are influenced by a range of internal and external factors, including destination characteristics, demographic and psychographic attributes. This two-stage approach provides a more understanding of tourist spending behavior and offers valuable things for designing targeted marketing and tourism development strategies [5].

Visit with us tourism company aims to expand its customer base by launching a new travel package, to reach the right customer target and to improve marketing strategies. Because the company currently offers five types of packages: Basic, Standard, Deluxe, Super Deluxe, King, only 18 % of customers purchased the packages in the last year due to random customer targeting. C. Mitchell created a system that can help this companies predict which customers will be targeted for new product offerings in order to increase conversion rates and reduce marketing costs by analyzing the socio-demographic and economic factors including age, income, and travel behavior and past interactions. The travel dataset is used from Kaggle that includes 4888 rows and 20 columns (Travel.csv) in this system [2].

3. System Design and Methodology

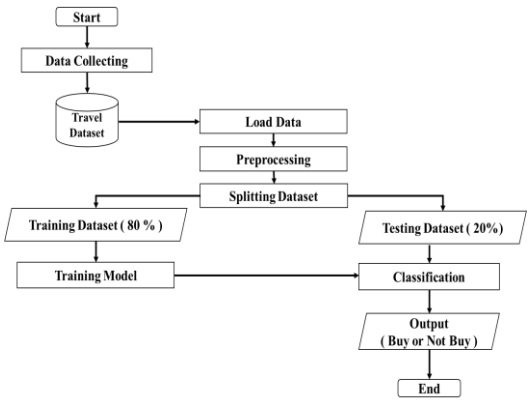


Figure 1. Proposed system design

3.1. Proposed System Design

The proposed Travel Package Purchase Prediction System is built to predict "the customers buy or not a travel package" by using Random Forest and XGBoost Classification algorithms. The workflow of this system is demonstrated in Figure 1 are:

- *Data Collecting*: Data is collected from users through Google Forms and paper-based surveys.
- *Travel Dataset*: The collected data is stored into a structured dataset.
- *Load Data*: The stored dataset is loaded into the system for further processing.
- *Preprocessing*: This step involves cleaning and preparing the data to ensure the dataset is ready for training.
- *Splitting Dataset*: The cleaned dataset is split into training dataset (80%) is used to train models and testing dataset (20%) is used for evaluating model performance.

3.2. Dataset

The dataset consists of 12 categorical features including target value that are described in Table 1. The dataset has 500 samples in this system. The data is collected using both online surveys via Google Forms and printed questionnaires distributed directly to individuals to local people in Myanmar. In the survey questionnaires form involves three parts

i. *Personal Information*: In this part, there are six questions to know the demographic of participants.
ii. *Previous Travel and Purchase Information*: In this section, five questions are used to know this information.
iii. *Preferred Travel Style*: There are three questions are questioned to know about the future travel style of participants.

Table 1. Features Description

No	Features	No	Features
1.	CustomerID	7.	Monthly Income
2.	Age	8.	No: of Person Visiting
3.	Gender	9.	No: of Trips in a Year
4.	Marital Status	10.	Package Name
5.	Occupation	11.	Preferred Package Cost
6.	Designation	12.	Package Buy/ Not Buy

3.3. Data Preprocessing

Data preprocessing is an essential foundation step in ML because the quality of input data directly affects the performance and accuracy of models. In this system, the data preprocessing stage is completed by the following steps:

- *Handling Missing Values:* Many data imputation methods can be used to fill the missing data depend on the data types of features. In this system, missing values are imputed using the most frequent value for each feature to preserve data integrity and avoid bias.
- *Removing Duplicate Rows*: Duplicated rows are identified and removed to ensure data quality and prevent redundancy in model training.
- *Encoding Categorical Features:* Categorical features are transformed into numerical format using appropriate encoding techniques.

3.3. Classification Models

3.3.1. Ensemble Learning Algorithms

It is a combination of two or more weak machine learning models to produce better predictions. There are two types of ensemble models:
(i) *Parallel Ensemble Methods*: It trains each base learner in parallel and independent form each other's. It can be divided into homogenous parallel methods (Bagging) and heterogeneous parallel ensembles (Stacking). Random forest follows the homogenous ensembles because it uses the same base algorithms.
(ii) *Sequential Ensemble Methods*: It trains a new base learner on the misclassifying samples in the dataset made by the previous one to minimize the errors. XGBoost follows the boosting algorithms of sequential methods [6].

3.3.2. Random Forest Algorithm

Random Forest (RF) is one of the ensemble ML algorithms. It can be used for classification and regression problems in ML. The base estimators in random forest are decision trees. Random forest randomly selects a set of features to decide the best split at each node of the decision tree. Finally, it takes each prediction from multiple decision trees and predicts the final output based on the majority votes of predictions. The overall workflow of proposed Random Forest Classification is shown in Figure 3.

In RF calculation involves five steps as shown in Figure 4. There are 12 features in own travel dataset including the target value. In step 1, the number of features for each bootstrap dataset is calculated the square root of total number of features in the whole dataset. Therefore, the number of features is 3 in each bootstrap dataset without including target value.

$$Number\ of\ Features\ = \sqrt{12}\ = 3.14$$

In step 2, the Gini index which is a measure of how impure a dataset for the whole bootstrap dataset and each feature of bootstrap dataset is calculated. Then the minimum value of Gini index is selected to define the root node and split the tree among the Gini index of features. The example shown in Figure 2.

$$Gini_{Index\,(T)} = 1 - \sum_{i=1}^{m} P_i^2 \qquad (1)$$

$$Gini - Index(T) = 1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2 = 0.5$$

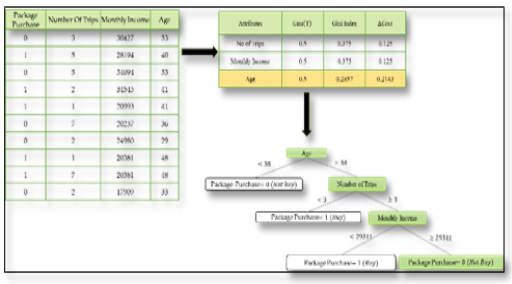


Figure 2: Example of generating a bagging tree in RF

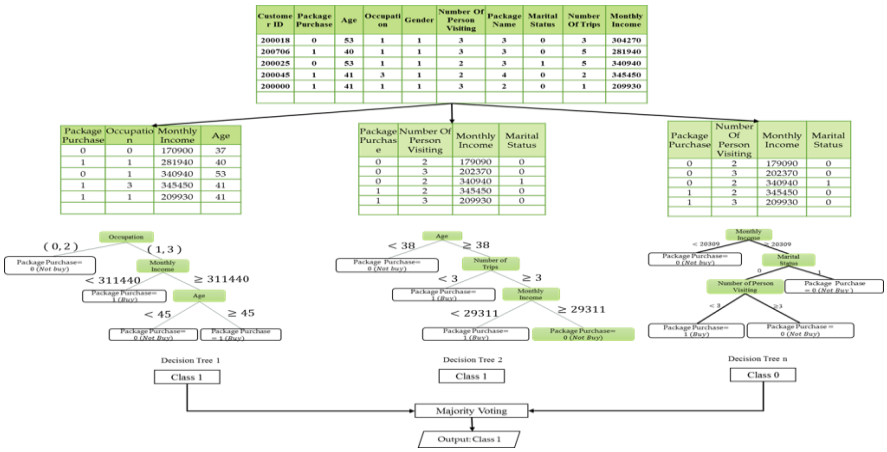


Figure 3. Workflow of random forest algorithm

Random Forest Algorithm Steps

Step 1: Select random data points (Bootstrap Dataset) from the original dataset and define the number of features for each selected bootstrap dataset.

$$Number\ of\ Features\ = \sqrt{Total\ Number\ of\ Features}$$

Step 2: Build the CART decision trees using Gini Index and ΔGini(Attribute) to define the best tree splitting subset with the minimum Gini Index value for each bootstrap sample dataset.

$$Gini_{Index(T)} = 1 - \sum_{i=1}^{m} P_i^2$$

$$\sum P_i^2 = \left(\frac{Yes}{TotalNoOfSamples}\right)^2 - \left(\frac{No}{TotalNoOfSamples}\right)^2$$

$$\Delta Gini(Attribute) = Gini(T)\ - Gini(T, Attribute)$$

Step 3: Each Decision tree generate an output by analyzing the incoming features as shown in Figure 3.

Step 4: Repeat Step 1, 2 and 3.

Step 5: Determine the final output by combining the results of all decision trees through the majority voting as shown in Figure 2.

Figure 4. Workflow of proposed random forest classifier algorithm

3.3.3. Extreme Gradient Boosting Algorithm

Extreme Gradient Boosting (XGBoost) is an advanced implementation of Gradient Boosting designed for high performance and efficiency in machine learning tasks. It is widely used for supervised learning problems, including Classification and Regression. It is an ensemble machine learning algorithm that follows the boosting technique. It combines the prediction of multiple weak learners to form a stronger prediction [9].

XGBoost calculation involves five steps as shown in Figure 5. In step 1, the initial prediction value is 0.5 because the proposed system has two target values buy or not.

In step 2, the residual values for each sample in the training dataset is calculated by substituting the predictive value from the actual value as shown in Figure 6.

In step 3, the Similarity scores is calculated to find the best splitting for each feature. And then Gain of each split is calculated and selected the highest value to create a boosting tree. This tree generates the output values. The learning rate is need to consider to get the probability of the generated output values.

In step 4, the initial prediction is converted to the *log (odd)* value and perform the step of adding the output value multiplied by learning rate to get the *log (odd)* prediction of output value.

Extreme Gradient Boosting Algorithm Steps

Step 1: Take the initial predictions for given input features and target feature start with a default value with the average value of all the target values in the dataset.

Step 2: Calculate the residuals value that is the difference between actual and predictive value to build XGBoost decision tree.

$$Residuals = Observed\ Value\ - Predictive\ Value$$

Step 3: Calculate the Similarity scores, Gain and Output values to build XGBoost Decision Trees.

$$SimilarityScore = \frac{(\sum Residual_i)^2}{\sum[P(Previous)_i \times (1 - P(Previous)_i)] + \lambda}$$

$$Gain = Left_{Similarity} + Right_{Similarity} - Root_{Similarity}$$

$$Output\ Values = \frac{\sum Residual}{\sum\left[P(Previous) \times \left(1 - P(Previous)\right)\right] + \lambda}$$

Step 4: Make a new prediction by calculating the new probability.

- Firstly, convert the initial prediction into log(odd).

$$log_2\left(\frac{P}{1-P}\right) = log_2(odds)$$

- Perform the step of adding the output node multiplied by learning rate.

$$log(odds)\ Predict = log(odds) + (Learning\ Rate\ (\varepsilon) \times Output)$$

- Convert the log(odd) values to the probability

$$Probability = \frac{e^{log\ (odds)\ Prediction}}{1 + e^{log\ (odds)\ Prediction}}$$

Step 5: Repeat Step 2 through Step 4 until the residuals are sufficiently small.

Figure 5. Workflow of proposed XGBoost classifier algorithm

Finally, the final prediction of first boosting tree is generated as shown Figure 6. The incorrect prediction of samples of the first tree is the training dataset of the next tree to update and improve the predictions. This process is repeated until the residuals are sufficiently small.

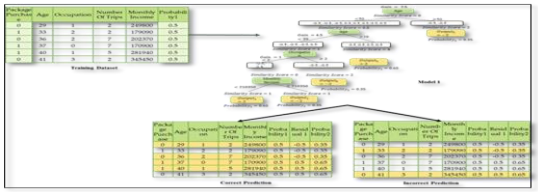


Figure 6. Example of generating a boosting tree in XGBoost

3.3.4. Decision Tree

A decision tree is a powerful machine learning algorithm used for classification and regression tasks. This model makes decision-making processes using a tree-like structure of nodes. The workflow of a decision tree starts by selecting the best feature to split the dataset at the root node. And then the data is repeatedly split into subsets until a stopping condition is met. The final leaf nodes represent the output class. Decision trees are easy to implement and perform on both numerical and categorical data [7],[8].

3.3.5. Logistic Regression

Logistic regression is a widely used one of the classification algorithms that produce the probability of a binary or multiclass outcome using a logistic sigmoid function. The workflow of logistic regression starts with data preprocessing, and scaling numeric features if needed. And then the model is training, where the algorithm fits the data by predicting coefficients that minimize the bias using a cost function like log loss. The sigmoid function is applied to the linear combination of features and weights to produce a probability score and then to convert the probability into a class label using a decision threshold.

3.3.6. Hyperparameter Tuning

Hyperparameter tuning is very important to improve the accuracy and performance of machine learning models. There are many hyperparameter tuning methods such as grid search, random search, Bayesian Optimization. In this system, Random Search and Grid Search are used to find the better hyperparameters for the Random Forest and XGBoost models. Random Search algorithm randomly selects combinations of hyperparameters to find the best one instead of checking every possible situation. Grid Search systematically selects all possible combinations of hyperparameters from a predefined set.

4. Results and Discussion

This section presents the detailed summary of the outcomes of the model implementation, compares them with related studies and discusses limitations.

The proposed system is developed using ensemble learning techniques, specifically Random Forest and XGBoost, to identify the customers based on demographic and behavioral data.

The results show that XGBoost and Random Forest achieved the accuracy of 97 % which is higher than models in related studies using similar classification models and the same number of samples in dataset.

The proposed system faced limitations due to the imbalanced the dataset and its relatively small size of 500 samples. And Kalaw is described as a start destination in five packages that are only used to collect the data from participants. These limitations could lead to bias the system prediction. To address this, future work should consider collecting more data and balanced the dataset and applying the hyperparameter tuning to improve the model robustness.

4.1. Accuracy Metrics

The proposed Random Forest and XGBoost models are evaluated using the following metrics:

- *Accuracy*: Measures overall correctness.
- *Precision*: Measures how many predicted positives are truly positive.
- *Recall*: Measures how many actual positives are correctly identified.
- *F1 score*: The harmonic meaning of precision and recall.
- *ROC-AUC*: Measures how well the model classified the two classes.

Table 2. Accuracy metrics of proposed system

Algorithms	Accuracy Metrics (%)				
	Accuracy	Precision	Recall	F1 Score	ROC_AUC
Random Forest	0.97	0.97	0.91	0.97	0.99
XGBoost	0.97	0.93	0.84	0.92	0.99

4.2. Accuracy Comparison of Proposed System with Different Number of Dataset

The robustness of the proposed system, an accuracy comparison is evaluated using various sizes of the dataset. The dataset is divided into five subsets containing different numbers of samples 50, 100, 200, and 500 to find how the model performance changed with data availability. The accuracy of the model improved with an increase in dataset size is described in Table 3 and demonstrated with bar chart in Figure 7.

Table 3. System accuracy with different number of datasets

Dataset	**Accuracy of Each Method**			
	Random Forest	**XG Boost**	**Logistic Regression**	**Decision Tree**
50	0.70	0.60	0.91	0.97
100	0.90	0.96	0.84	0.92
200	0.86	0.97	0.81	0.94
500	0.97	0.97	0.91	0.97

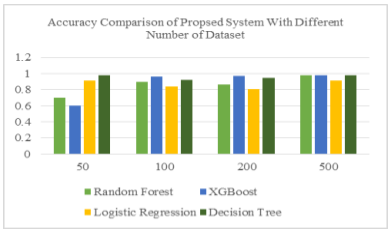


Figure 7. Bar chart for accuracy comparison of proposed system across the different dataset size

4.3. Accuracy of Proposed System Using SMOTE Data Balancing Technique

Synthetic Minority Oversampling Technique (SMOTE) is designed to solve imbalanced datasets by creating the virtual samples for the minority class. The imbalanced dataset of this sytsem is balanced by using SMOTE that could be captured the important features of the minority class and reduced the bias. In this system, SMOTE is used to make a better performance of classifier models [1], [6]. The results after using data balancing technique are demonstrated with table in Table 4 and bar chart in Figure 8.

Table 4. Accuracy metrics using SMOTE

Algorithms	Accuracy Metrics (%)				
	Accuracy	Precision	Recall	F1 Score	ROC_AUC
Random Forest	0.98	0.94	0.94	0.98	0.96
XGBoost	0.97	0.93	0.88	0.97	0.93

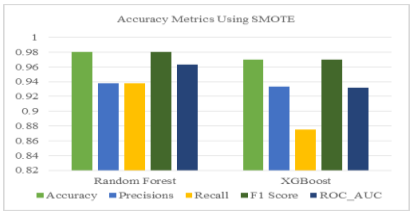


Figure 8. Accuracy metrics using SMOTE

4.4. Accuracy Comparisons of Proposed System Using Hyperparameter Tuning

The proposed system results after using hyperparameter tuning are demonstrated in Table 5.

Table 5. Accuracy metrics using hyperparameters

Hyperparameter Tuning Methods	Accuracy	
	Random Forest	XGBoost
Random Search	0.97	0.97
Grid Search	0.97	0.99

4. Conclusions and Future Work

The proposed travel package prediction system is developed using Random Forest and XGBoost algorithms to classify the travel package information that is potential customers based on their demographic and behavioral data. The accuracy of XGBoost model is 97% and the accuracy of Random Forest is 97% on testing data.

The system enables the travel company to move away from random customer targeting toward a data-driven approach that improves marketing efficiency. Travel agencies could be developed more personalized, attractive offerings, leading to greater business success in Southern Shan State by accurately predicting customer behavior.

The future work should be considered collecting more data and balanced datasets, analyzing for new packages with famous places in Myanmar. Moreover, a new system could be built the proposed prediction system with combining tourism destination recommender system.

Acknowledgements

I would like to be thankful to everyone who helped me to finish this study. There are so many people who I want to thank. Firstly, I would like to thank my parents for always supporting me financially and emotionally.

And then, I would also like special thanks to gratitude our teacher, Dr. Nan Saw Kalayar, Professor, University of Computer Studies (Taunggyi), Shan State, for sharing the professional knowledge in different areas. And she gave the guideline and the start point for preparing the master thesis.

References

[1] A. Parmar, “Overcoming Class Imbalance using SMOTE Techniques,” Analytics Vidhya, October, 2020.

[2] C. Mitchell, “Ensemble Techniqes Travel Package Purchase Prediction”, December 2022.

[3] I. Pinto “Online travel agencies: Factors influencing tourist purchase decision,” *Journal of Tourism, Hospitality and Environment Management*, vol. 4, no. 13, pp. 12–22, 2019.

[4] J.C Sancho Nunez, “Machine learning applied to tourism: A systematic review”, May 2024.

[5] J.L. Nicolau and F.J. Mas, "Heckit modelling of tourist expenditure: evidence from Spain", International Journal of Service Industry Management, Vol. 16 No. 3, pp, 2005.

[6] M. N. F. M. Noor, M. F. M. Faudzi, M. H. Hassan, M. A. M. Yunus, and A. T. S. Saufi, “XGBoost and Random Forest Optimization using SMOTE to Classify Air Quality,” ResearchGate, May 20, 2024.

[7] Murel, J., PhD, & Kavlakoglu, E. (2025, April 16). Ensemble learning. What is ensemble learning?

[8] R. C. Joshi and P. Bansal, “A Comparative Study of Decision Tree ID3 and C4.5,” SAI Conference, Special Issue No. 10, pp. 12–15, March 10, 2018.

[9] T.Chen and C.Guestrin, “XGBoost: A Scalable Tree Boosting System”, 16, August,2016

Measuring Customers' Satisfaction on Restaurant Reviews Using Support Vector Machine

April Aung
Nan Saw Kalayar
University of Computer Studies Taunggyi
aprilaung@ucstgi.edu.mm
sawkalayar@ucstgi.edu.mm

Abstract

Nowadays, online reviews or customer reviews have become very important for the success of businesses. This paper proposes a sentiment classification system designed to accurately measure customer satisfaction from Myanmar restaurant reviews by using Support Vector Machine (SVM). First, the process begins with data collection and organized the collected data into a dataset. Then, preprocessing steps are applied to enhance processing efficiency and improve the accuracy of the analysis. For feature extraction the system uses two Algorithms, the first one is a Term Frequency, Inverse Document Frequency (TF-IDF) and the second one is Bidirectional Encoder Representation from Transformer (BERT). After that, split the dataset for training, testing used and train the data. The testing data is match with the data from train model and output the label "satisfied (1)" or "unsatisfied (-1)". There are two datasets used in this system. The first dataset is "restaurant-reviews" dataset from Kaggle which include 7828 data. The second dataset is created by own collection Myanmar restaurant reviews data and give the dataset name "Own Restaurant Reviews Dataset" which include 5000 data. Own data are collected from the websites TripAdvisor, Facebook, YouTube and TikTok. Hybrid dataset ("restaurant-reviews" dataset + Own Restaurant Reviews dataset) is created and there are all total 12828 reviews data in this dataset. Using BERT model with the SVM classifier, the model achieves 90% accuracy and shows a clear improvement in performance. This highlights the effectiveness of contextual embedding over traditional feature extraction techniques in sentiment analysis tasks.

1. Introduction

In the digital era, online reviews have become an important source of information for both consumers and businesses. Customers often share their dining experiences on social media and review platforms. Analyzing these reviews using sentiment analysis which is a subfield of Natural Language Processing (NLP), this helps identify opinions and emotions expressed in text. By determining whether a review expresses a positive or negative sentiment, businesses can evaluate the level of customer satisfaction and make informed improvements to their overall quality.

This study uses a dataset which consist of 12828 customer reviews data. Among them 5000 data are own collections restaurants reviews data and 7,828 are "restaurant-review" data from Kaggle. To make the data clean and consistent, the system performs several preprocessing steps. There are two feature extraction model are used in this system. The first one is Term Frequency - Inverse Document Frequency (TF-IDF) and the second one is Bidirectional Encoder Representations from Transformers (BERT). This system focuses on developing a sentiment analysis system that classify customer reviews into two categories: satisfied (1) and unsatisfied (-1).

This system uses the popular machine learning model Support Vector Machine (SVM) to measure customer satisfaction. The system achieves an accuracy of 78% with SVM. However, with the emergence of deep learning and contextual embeddings, the need for more advanced models has become apparent. The main goal of this system is to classify restaurant reviews as either satisfied or unsatisfied. It aims to build an accurate sentiment analysis model to understand customer opinions.

2. Related Work

L. Gunawan, M. S. Anggreainya, L. W. Santya, G. Y. Lesmanaa, and S. Yusufa conducted an emotional sentiment analysis on restaurant reviews using the Support Vector Machine (SVM) algorithm. The dataset comprised restaurant reviews specifically from the TripAdvisor website, limited to establishments located in Jakarta, Indonesia. Among the classifiers evaluated, SVM achieved the highest accuracy at 0.79, followed by Naïve Bayes with an accuracy of 0.77 [6].

T. Ahmed Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. B. M. Su'ud conducted a sentiment analysis using both the Support Vector Machine (SVM) and Random Forest algorithms. The results demonstrated that SVM slightly outperformed Random Forest, achieving an accuracy of 0.80394 compared to 0.78564 [11].

S. H. Imanuddin, K. Adi, and R. Gernowo conducted sentiment analysis on the Satusehat mobile application using the Support Vector Machine (SVM) algorithm. Their study utilized a dataset of 25,000 user reviews, comprising 18,359 negative and 6,641 positive instances. The focus was on classifying user feedback into sentiment categories, where the SVM classifier achieved a high accuracy of 91% [10].

P. H. Prastyo, A. S. Sumi, A. W. Dian, and A. E. Permanasari conducted a sentiment analysis study on public reactions to the Indonesian government's handling of COVID-19, utilizing Twitter data and the Support Vector Machine (SVM) algorithm with a Normalized Polynomial Kernel. Tweets were collected using the Twitter scraper library and labeled into positive, negative, and neutral. The SVM model demonstrated superior performance in binary sentiment classification with an average accuracy of 82.00% [7].

M. O. Pratama, W. Satyawan, R. Jannati, B. Pamungkas, Raspiani, M. E. Syahputra, and I. Neforawati conducted sentiment analysis on public opinions regarding Indonesia's Commuter Line (KRL) services using Twitter data. The authors applied three machine learning classifiers Multinomial Naive Bayes (MNB), Random Forest (RF), and Support Vector Machine (SVM) to the preprocessed data. Among the models tested, SVM achieved the highest classification accuracy at 85%, indicating its strong capability in identifying sentiment patterns. [5].

3. System Design and Methodology

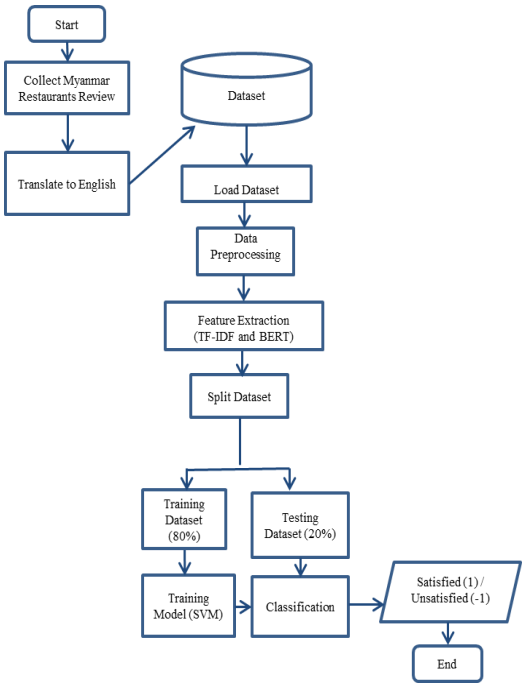


Figure 1. Proposed system design

The proposed system begins with collecting customer reviews data for Myanmar restaurants from Online sources. The dataset, comprising 12,828 reviews. This data is passed to the preprocessing steps. After preprocessing, Features are extracted using TF-IDF. Additionally, this system uses BERT model to make feature extraction process for each review. The dataset is split into training (80%) and testing (20%). To test the data, the testing data is matching with the data from training model and output the label result "Satisfied (1)" or "Unsatisfied (-1)".

3.1. Dataset

In this system three datasets are used, which are "Own Restaurant Reviews" dataset, "restaurant-reviews" dataset from Kaggle and Hybrid dataset ("restaurant-reviews" dataset + Own Restaurant Reviews dataset).

"Own Restaurant Reviews" dataset consist of 5000 reviews and it is collected from Trip Advisor and most are from social media Myanmar restaurant pages such as Facebook, TikTok, YouTube, and food blogger blogs. "Own Restaurant Reviews" data are collected with both English and Myanmar languages. The review with Myanmar Language is translated to English to increase language consistency. The dataset includes feedback from popular tourist dining spots and traditional Myanmar food restaurants. Traditional food items reviewed include well-known Myanmar dishes such as Mohinga, Shan Noodles, Laphet Thoke, Ohn No Khao Swè, Samosa Thoke, Nan Gyi Thoke, Mont Linmaya and Shwe Yin Aye. Additionally, the dataset features reviews of fast food such as KFC and Lotteria. [5].

The "restaurant-reviews" dataset from Kaggle is utilized in this study to evaluate sentiment classification performance. Originally containing 10,000 reviews, the dataset includes various fields such as restaurant name, reviewer, review text, rating, metadata, timestamp, and pictures. This system uses only the "review text" column from this dataset. Neutral reviews (21.72%) are removed during pre-processing to focus solely on clear positive and negative sentiments. After this filtering process, a total of 7,828 labeled reviews remained for training and evaluation [12].

The Hybrid dataset ("restaurant-reviews" dataset + Own Restaurant Reviews dataset) dataset is constructed. This combined dataset aims to improve the robustness and generalization capability of sentiment classification models. The final hybrid dataset comprises a total of 12,828 reviews, of which 7,789 are labeled as positive and 5039 as negative.

Table1. Dataset using in the proposed system

Dataset	Reviews
"Own Restaurant Reviews"	5000
"restaurant-reviews"	7828
("restaurant-reviews" dataset + Own Restaurant Reviews dataset)	12828

3.2. Preprocessing and Feature Extraction

In machine learning, preprocessing is important to speed up the system accuracy. In this system, first the restaurant review data are collected. Some reviews with Myanmar language are translate to English Language to ensure uniformity in text processing. This data is passed to the preprocessing steps.

After preprocessing step, two famous algorithms are applied for feature extraction: BERT and TF-IDF. The detail is explained in the following section. Feature extraction is the process of turning unprocessed text data into numerical formats that machine learning models could be used [6].

3.3. Bidirectional Encoder Representations from Transformers

In this study, BERT is used for feature extraction and combination this with a Support Vector Machine (SVM) classifier for sentiment analysis.

The BERT model was used to generate contextual word embeddings that capture semantic meaning based on surrounding context, enabling the understanding of complex language patterns [2]. In BERT, each review is processed using the pre-trained BERT model (Bert-base-uncased) via the Hugging Face Transformers library, where [CLS] and [SEP] tokens are automatically added as shown in Figure2. The 768-dimensional [CLS] token embedding is extracted to represent the entire review and used as input features for the SVM classifier.

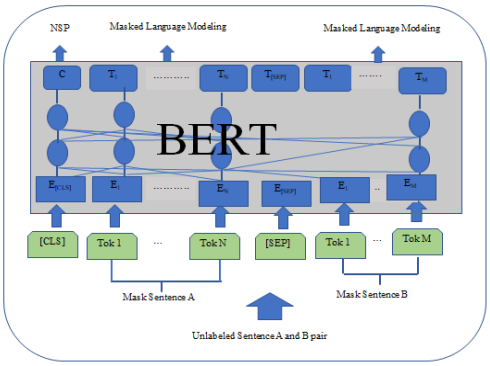


Figure 2. BERT Pre-trained Process

BERT is pre-trained using two main tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). These two tasks help BERT understand the meaning and relationships of words and sentences.

In MLM, some words in a sentence are randomly masked, and the model learns to predict the original words based on the context from both directions (left and right).

In NSP, the model is given pairs of sentences and learns to predict whether the second sentence logically follows the first one [13].

3.3.1 Input Embedding in BERT

There are three main components in input embedding process of BERT: token embeddings, segment embeddings, and positional embeddings as illustrated in Figure 3.

i. Token Embeddings

Token embeddings represent the meaning of individual words or sub words. In BERT, each input token (such as a word or part of a word) is mapped to a fixed-size vector using a pre-trained embedding table. This helps the model understand the semantic content of the tokens.

ii. Segment Embeddings

Segment embeddings help BERT differentiate between multiple sentences in a single input. BERT can handle sentence pairs (e.g., question and answer), and it assigns a segment ID (either 0 or 1) to each token to indicate whether it belongs to sentence A or sentence B.

iii. Positional Embeddings

Positional embeddings encode the position of each token in the input sequence. Since BERT does not use recurrence or convolution, positional embeddings are added to the token vectors to retain the order of the sequence. These embeddings are learned and fixed during training [13].

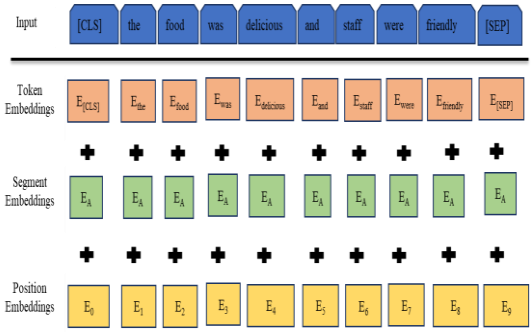


Figure 3. BERT Embedding process

3.4. Term Frequency -Inverse Document Frequency (TF-IDF)

TF-IDF (Term Frequency-Inverse Document Frequency) is a method used to change text into numbers by measuring how important each word is in a review. In this system, TF-IDF is used to turn customer reviews into useful features, which are then given to an SVM model to find out if the sentiment is positive or negative [4]. In TF-IDF, the input text data was cleaned and transformed through a comprehensive pipeline to enhance its quality and reduce noise [3].

The following steps are applied:

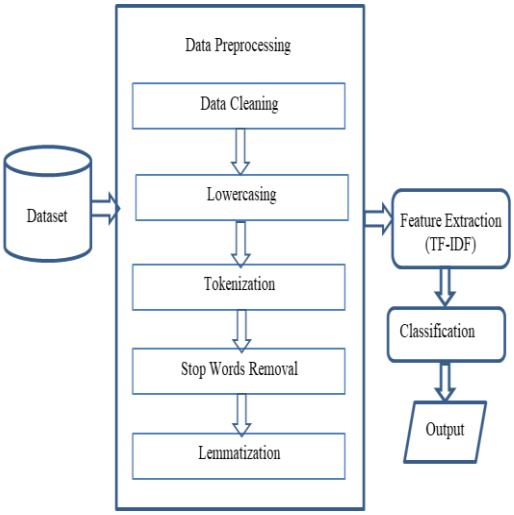


Figure 4. Block Diagram of Data preprocessing for TF-IDF

No.	Before Preprocessing	After Preprocessing
Review 1	The service was good and also food was good. Can also chill with friends.	[“service”,” good”,” food”, ” good”, “chill”, “friends”]
Review 2	Good food and good ambiance. Also, the music is amazing.	[“goof”, “food”, “good”, “ambiance”, “music”, “amaze”]

Figure 5. Input data before and after data preprocessing for SVM (TF-IDF)

Mathematically, TF-IDF for a term t in a document d is calculated as:

$$TF - IDF(t,d) = TF(t,d) * \log(\frac{N}{DF_{(t)}}) \quad (1)$$

Where:

- TF(t,d) is the number of times term *t* appears in document *d*,
- N is the total number of documents,
- DF (*t*) is the number of documents containing the term *t*.

This transformation converts each review into a high-dimensional sparse vector, with each dimension representing the TF-IDF weight of a word in the vocabulary. The resulting feature matrix is well-suited for use with the Support Vector Machine (SVM) classifier, allowing it to distinguish between satisfied and unsatisfied reviews based on word usage patterns [9].

3.5. Support Vector Machine

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm commonly used for classification and regression. It works by finding the optimal boundary, known as hyperplane, to separate data point. The four main types of SVM are Linear, Polynomial Kernel, RBF Kernel and sigmoid Kernel [8].

SVM operates by identifying the optimal hyperplane that best separates data points from different classes, maximizing the margin between the closest support vectors of each class. One of the key strengths of SVM is its ability to handle both linear and non-linear classification problems through the use of kernel functions [1].

In this study, a non-linear SVM was employed due to the nature of restaurant reviews, which often contain complex and context-dependent expressions. Sentiment in textual data is rarely determined by individual words alone but is influenced by phrases and contextual combinations phrases like “not bad” expresses a positive sentiment, whereas “not good” expresses a negative sentiment. Such subtleties necessitate a model capable of capturing non-linear patterns in high dimensional space [7][8][4].

To address this, the kernel trick was applied, transforming the input features into a higher-dimensional space where a linear separation becomes possible. Specifically, a polynomial kernel was used, as it is well-suited for datasets where the relationship between features and class labels is non-linear and follows a polynomial pattern.

4. Result and discussion

This section presents the experimental results of sentiment classification on restaurant reviews using two models: BERT and TF-IDF.

Table2. Accuracy comparison across different datasets

Dataset	TF-IDF	BERT
"Own Restaurant Reviews Dataset"	74.83%	82.45%
"restaurant-reviews" dataset	66.61%	73.58%
("restaurant-reviews" dataset + "Own Restaurant Reviews Dataset")	78%	90%

Table2. Shows the accuracy comparison between two feature extraction approaches: TF-IDF and BERT across three different dataset types: "Own Restaurant Reviews Dataset", "restaurant-reviews" dataset from Kaggle, and ("restaurant-reviews" dataset + Own Restaurant Reviews dataset). The results show that BERT consistently outperforms TF-IDF for all dataset types. For the "Own Restaurant Reviews Dataset", BERT-based features achieved an accuracy of 82.45%, compared to 74.83% with TF-IDF. In the Kaggle dataset, the accuracy increased from 66.61% TF-IDF to 73.58% BERT. Similarly, for the hybrid ("restaurant-reviews" dataset + Own Restaurant Reviews dataset), BERT yielded 78.12% accuracy, significantly higher than the 69.70% achieved using TF-IDF.

Table3. Comparison of hybrid dataset based on data volume

Dataset Size	TF-IDF	BERT
500	66.69%	76.36%
5000	72.20%	82.20%
12828	78%	89.62%

Table3 shows the comparison result of the accuracy of SVM (TF-IDF) and SVM (BERT) across three different dataset sizes: 500, 5,000, and 12,828 reviews. As the dataset size increases, both models improve in accuracy. SVM (BERT) consistently outperforms SVM (TF-IDF) at all sizes, achieving 76.36% vs. 66.69% at 500 reviews, 85.20% vs. 72.20% at 5,000 reviews, and 89.62% vs. 78% at 12,828 reviews.

Table4. Accuracy comparison of hybrid dataset (Imbalance vs. Balance)

Dataset Types	TF-IDF	BERT
Hybrid (Imbalance data)	78%	89.62%
Hybrid (Balance data)	79.20%	92.30%

Table4 compares the accuracy of SVM classifiers using TF-IDF and BERT on 12,828 restaurant reviews (7,789 positive, 5,039 negative). On the imbalanced dataset, TF-IDF scored 68.30% accuracy, while BERT reached 90.3%. After balancing the data (6,400 positive, 6,428 negative), accuracies improved to 78.45% for TF-IDF and 92.30% for BERT.

Table5. Accuracy results by polynomial kernel degree in SVM

Polynomial Kernel Degree	Accuracy (%)
d=1	79.6%
d=2	78%
d=3	79.20%

The degree (d) of the polynomial kernel and the regularization parameter (C) are important hyper parameters in SVM. Degree d controls the complexity of the decision boundary. In this study, d was tested from 1 to 3, with accuracies of 84.6% (d=1), 83.9% (d=2), and 84.2% (d=3), as shown in Table 3. Although d=1 gave the highest accuracy, it represents a linear model, while d=2 provided a better balance between performance and complexity. The regularization parameter C was set to 1, offering a fair trade-off between fitting the data and avoiding over fitting.

Table6. Epoch accuracy

Epoch	Accuracy (%)
1	70.5%
3	81.8%
5	87.21%
7	89.5%
8	90.3%
10	90.3%

An epoch is one complete pass through the training data in BERT-based sentiment analysis. During each epoch, the model learns by processing all samples and adjusting its weights to reduce errors. This repeated learning helps the model improve its performance over time.

Table6 show the model's progress over 10 epochs. Accuracy started at 70.5% in epoch 1 and rose to 81.8% by epoch 3. By epoch 5, it reached 87.21%, showing effective learning. Accuracy continued to improve, hitting 89.5% at epoch 7 and 90.3% at epoch 8. It remained stable at 90.3% through epoch 10, indicating the model had likely reached its peak performance.

Table7 shows the performance comparison between SVM models using TF-IDF and BERT embeddings. The TF-IDF model performed poorly on negative reviews, with a low recall of 0.16 and an F1-score of 0.27 for class -1. However, it performed better on positive reviews, achieving an F1-score of 0.78. In contrast, the BERT model showed strong and balanced results for both classes, with F1-scores of 0.94 for class -1 and 0.96 for class 1, proving its better effectiveness in sentiment classification.

Table7. Evaluations

Model	Label	Precision	Recall	F1-Score
TF-IDF	-1	0.91	0.16	0.27
	1	0.65	0.99	0.78
BERT	-1	0.94	0.93	0.94
	1	0.96	0.96	0.96

5. Conclusion

This study presents an effective sentiment analysis system for measuring customer satisfaction based on restaurant reviews. By using both traditional TF-IDF and advanced BERT for features extraction and combined with the Support Vector Machine (SVM) classifier, the system successfully categorizes customer feedback into satisfied and unsatisfied sentiments. This demonstrates the effectiveness of using modern language models like BERT alongside traditional classifiers to enhance sentiment analysis in real-world applications.

For Further Extension, more restaurant reviews data will be collected and use this dataset with other machine learning model.

References

[1] A. Alqurafi and T. Alsanoosy, "Measuring Customers' Satisfaction Using Sentiment Analysis: Model and Tool," Department of Computer Science, College of Computer Science and Engineering, Taibah University, Saudi Arabia, 7 February 2024, pp. 419-430.

[2] B. Rahman and M. Maryani, "Optimizing Customer Satisfaction Through Sentiment Analysis: A BERT-Based Machine Learning Approach to Extract Insights," Nasional University and Bina Nusantara University, Jakarta, Indonesia, 6 October 2023.

[3] E. Hossain, O. Sharif, M. M. Hoque, and I. H. Sarker, "SentiLSTM: A Deep Learning Approach for Sentiment Analysis of Restaurant Reviews," Chittagong University of Engineering and Technology, Bangladesh, 17 May 2021.

[4] K. Zahoor, N. Bawany, and S. Hamid, "Sentiment Analysis and Classification of Restaurant Reviews Using Machine Learning," DOI: 10.1109/ACIT50332, 6 January 2021.

[5] M. O. Pratama, W. Satyawan, R. Jannati, B. Pamungkas, Raspiani, M. E. Syahputra, and I. Neforawati, "Sentiment Analysis of Public Opinions on KRL Services Using Twitter and Machine Learning Algorithms," Faculty of Computer Science, University Pakuan, Indonesia, 2022, pp. 55–62.

[6] L. Gunawan, M. S. Anggreainya, L. W. Santya, G. Y. Lesmanaa, and S. Yusufa, "Support Vector Machine Based Emotional Analysis of Restaurant," Computer Science Department, School of Computer Science Bina Nusantara University, Jakarta, Indonesia, December 2022.

[7] P. Devi Durga, G. Harani Shree, and S. Ranjani, "Restaurant Review Using Sentiment Analysis in Social Media," ISSN-2349-5162, May 2023.

[8] P. H. Prastyo, A. S. Sumi, A. W. Dian, and A. E. Permanasari, "Sentiment Analysis of Public Opinion on the Indonesian Government's COVID-19 Response Using Twitter Data and Support Vector Machine," Faculty of Computer Science, Universitas Indonesia, Indonesia, 2021, pp. 102–110.

[9] R. Atre and N. Tapaswi, "A Prediction of Customer Behavior Using Logistic Regression Algorithm," International Journal of Computer Applications (0975 – 8887) Volume 183 – No. 50, February 2022.

[10] S. H. Imanuddin, K. Adi, and R. Gernowo, "Sentiment Analysis on Satusehat Application Using Support Vector Machine Method," Vol. 5, No. 3, July 2023, pp. 143–149, eISSN: 2656-8632.

[11] T. Ahmed Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. B. M. Su'ud, "Sentiment Analysis uses Support Vector Machine and Random Forest," eISSN 2821-370X, February 2024.

[12] https://www.kaggle.com/datasets/joebeachcapital/restaurant-reviews

[13] https://towardsdatascience.com/a-complete-guide-to-bert-with-code-9f87602e4a11/

Ingredient-based Recipe Recommendation System using Latent Dirichlet Allocation (LDA)

Ei Pyae Phyo Khin
University of Computer Studies(Taunggyi)
eipyaephyokhin@ucstgi.edu.mm

Soe Soe Lwin
University of Computer Studies(Taunggyi)
soesoelwin@ucstgi.edu.mm

Abstract

Recipe recommendation systems have gained significant attention due to the rising interest in home cooking and meal planning. With the rapid growth of online recipe platforms, finding new dishes that suit their taste and available ingredients has become extremely challenging for users. Most recipe recommendation systems rely on collaborative filtering based on items' general popularity. This paper describes the content-based recipe recommendation system that utilizes Latent Dirichlet Allocation (LDA) and Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction to uncover semantic themes from a recipe corpus. The system employs a content-based filtering approach where recipes are represented by their ingredient lists. Term Frequency-Inverse Document Frequency (TF-IDF) is utilized to weight the importance of individual ingredients within recipes. Then, LDA is applied to extract latent topics (ingredient combinations) from the recipe corpus, allowing for a deeper understanding of underlying culinary patterns. Cosine similarity is then calculated between a user's input ingredients and the ingredient-topic distributions of recipes to generate the top-ranked recipes for recommendations. Experimental evaluation demonstrates that the hybrid approach significantly improves the relevance and diversity of recipe suggestions than the pure LDA approach.

Keywords—*recipe recommendation, content-based filtering, Latent Dirichlet Allocation, TF-IDF, topic modeling, cosine similarity*

1. Introduction

Recommendation systems have modified various industries, from e-commerce to entertainment, by using data to predict user preferences and deliver relevant content [8]. Many popular platforms including Amazon, YouTube, and Netflix are using recommendation algorithms to enhance user satisfaction. In the culinary domain, recipe recommendation systems can offer significant suggestions to users to avoid food waste by using currently available ingredients, and can promote culinary diversity by introducing various cuisines from different corners of the world. As many food platforms grow in popularity, users face information overload with finding desired recipes from large databases.

This paper presents an ingredient-based recipe recommendation system using Content-based Filtering (CB) and Latent Dirichlet Allocation (LDA) topic modeling approaches. Unlike Collaborative Filtering (CF), which relies on extensive user data such as ratings and reviews [8], the CB approach relies on item characteristics, specifically ingredients, to generate recommendations. LDA is a generative probabilistic model that identifies hidden patterns or topics within a recipe corpus, capturing the semantic relationships between them [3][4]. TF-IDF weights ingredient importance, enabling precise similarity calculations for recommendation.

The publicly available Food.com recipes dataset and manually collects Myanmar recipes are used to ensure culinary diversity, aiming to improve recommendations for underrepresented cuisines. Myanmar recipes written in English are often poorly structured and required manual preprocessing to address inconsistencies. The system accepts user input ingredients, applied topic modeling to both the input and the recipes, and ranks results using a combination of topic similarity and TF-IDF-based text similarity, allowing interpretable and flexible recommendations.

The paper is organized as follows: Section II reviews related work on recipe recommendation and topic modeling. Section III describes the methodology. Section IV presents data preprocessing, model design, evaluation, and experimental results. Section V concludes with findings and future research directions.

2. Related Work

This section reviews prior work relevant to this system, which utilizes TF-IDF and LDA for ingredient-based recommendation.

Freyne and Berkovsky [1] developed a CF system based on user-recipe interactions. The authors proposed TF-IDF based framework to retrieve content, focusing on efficient text representation, enabling recipe matching and achieve reasonable personalization but requires extensive user data. Teng et al. [2] investigates text analysis methods for pattern recognition, applying statistical models to extract meaningful features from unstructured data. Their work indicates that the recipes are not just texts, the ingredients in the recipe have special relationship between them which indicates their uniqueness.

Latent Dirichlet Allocation (LDA) is widely used for topic modeling in text analysis. Blei et al [4] introduced LDA to uncover latent topics in document corpora. Ge et al [3] use tags collected from users to identify their preferred ingredients and features. They proposed the technology to find latent factors between user interactions and recipes to improve personalized recommendations. Chhipa [5] propose a content-based filtering approach that uses ingredient profiles to recommend recipes, cosine similarity to match user preferences. The model is based on TF-IDF with cosine similarity which allowed users to search recipes based on available ingredients and can also filtered out the unwanted ingredients. Khan et al [9] presented personalized, health-aware recipe recommendation system considering both user preferences and nutritional facts. They investigated Ensemble Topic Modeling based Feature Identification techniques for efficient user modeling and recipe recommendation. Their work emphasizes the importance of contextual information for capturing broader culinary themes.

3. Methodology

This section describes the methodology used to develop the system, combining TF-IDF for feature extraction, LDA for topic modeling and cosine similarity for ranking.

3.1. Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure that reflect how important a word (ingredient) is to a document (recipe) in a collection of documents (corpus). It is a numerical measure, helps determine how relevant a word is to a specific document by considering both the frequency of the word within the document and its rarity across the entire corpus.

Term Frequency (TF) is typically calculated as the number of times a word (t) appears in a document (d) divided by the total number of words in that document.

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

Inverse Document Frequency (IDF) is calculated as the logarithm of the total number of documents (N) in the collection (D) divided by the number of documents containing the specific word.

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (2)$$

Term Frequency-Inverse Document frequency can enhance the content-based system by weighting the term importance. TF-IDF in recipe recommendation computes the similarity scores based on ingredient lists, entirely focusing on ingredients matching [5]. Combining LDA and TF-IDF can improve the framework by modeling both latent patterns and ingredient significance. TF-IDF captures the importance of specific ingredients while LDA captures the broader context in which the ingredients appear. The combination of granular and thematic insights can help the model produces more effective recommendations.

3.2. Topic Modeling with Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA), a generative probabilistic model, is one of the most popular methods for performing topic modeling. It uncovers the hidden "topic" that explain the relationship of words in the text documents [4]. In the context of recipes, LDA can uncover latent ingredient topics or themes that represent the common grouping of ingredients. LDA treats each recipe as a document and its ingredients as words, then extracts distinct ingredient clusters. Therefore, instead of relying on individual ingredients, recipes can be represented as distribution over latent topics. These topic distributions can serve as effective feature vectors for comparing and recommending recipes. If a user

has ingredient aligning strongly with the "Breakfast" topic, the system can recommend recipes that fit the theme, even if they don't have the exact specified ingredients. LDA based models are particularly suitable when there are limited interaction data but textual of semi-structured information is available [4].

LDA will be applied to the preprocessed ingredient corpus to discover latent culinary topics. It uses bag-of-words (BOW) representation to analyze the text data, therefore there is no syntax rule. LDA assumes that documents with similar topics will use a similar group of words. The model only needs to know the number of topics (K) to construct by Dirichlet distributions.

Plate notation for LDA uses boxes (plates) to represent the repeating structures in the model. The outer plate represents the collection of documents (M), and the inner plate represents the words within each document (D). Each plate can be viewed as a "loop", whereas the variable shown in the bottom right corner of the plate represent the number of iterations of the loop.

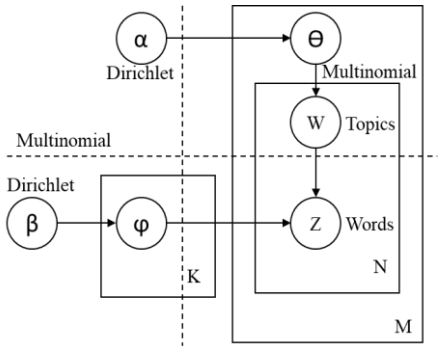


Figure 1. Plate notation for LDA with Dirichlet-distributed topic-word distributions

Variables representation within the plates:

- α: Dirichlet prior parameter for the topic distribution of each document
- β: Dirichlet prior parameter for the word distribution of each topic
- Θ: Topic distribution for each document M (sampled from Dirichlet α)
- φ: Word distribution for each topic K (sampled from Dirichlet β)
- Z: Topic assignment for each word in each document (sampled from Multinomial Θ)
- W: Observed word in each document (sampled from Multinomial φ)
- K: Number of topics

The generative process where LDA assumes that new documents are created in the following way. First, it determines the number of words in each document, and choose a topic mixture for the document over a fixed set of topics. Each word in the document is generated by picking a topic based on the document's multinomial distribution (Θ) and picking a word based on the topic's multinomial distribution (φ). LDA only need to know the number of topics (K) to construct and it tries to learn the topic representation of K topics in each document and the word distribution of each topic over a corpus. It tries to figure out the "pattern" for how each document could have been created.

LDA randomly assign each word in each document to one of the **K** topics. For each document **d**, LDA assume that all topic assignments expect for the current one are correct and calculate the two proportions. First it calculates the proportion of words in document d that are currently assigned to topic t = p (topic t | document d). And then calculate the proportion of assignments to topic t over all documents that come from this word w = p (word w | topic t). Multiply those two proportions and assign **w** a new topic based on that probability. Repeat this calculation until the assignments make sense, the steady state. Based on that output, similar documents can be identified within the corpus.

3.3. Cosine Similarity

Cosine similarity is the measure of similarity between two vectors, the dot product of the vectors divided by the product of their magnitude.

$$(\mathrm{A,B}) = \frac{\mathrm{A}\cdot\mathrm{B}}{\|A\|\cdot\|B\|} \qquad (3)$$

It value ranges from -1 to 1, the value closer to 1 indicates the higher similarity between the two vectors and closer to -1 indicate the higher dissimilarity. Cosine similarity is known for its simplicity and efficiency, performs well in high-dimensional spaces and can effectively handle sparse data, which is common in text-based or item-feature representation. Cosine similarity only considers the orientation (angle) between vectors which makes it suitable for situations where the frequency of words or features may differ greatly, it can capture their semantic or characteristic relatedness by the direction of their respective vectors.

The system utilizes the cosine similarity to determine the semantic similarity between the user's input ingredients list and the feature representation of each recipe in our dataset. The cosine similarity score was calculated between the user query vector and every recipe vector. Recipe with higher cosine similarity scores were considered more relevant and ranked higher in the recommendation list.

4. Experimental Results

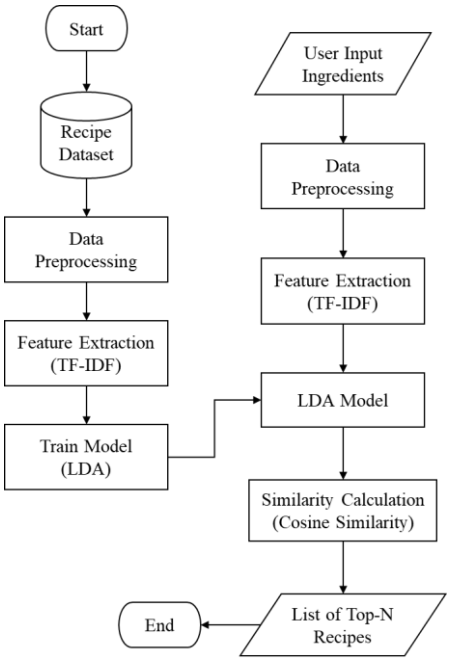


Figure 2. Proposed System Flow Chart

In figure 2, the system takes the user's input of available ingredients list and returns a ranked list of Top-N recommended recipes. All the recipes in dataset and the input ingredients are preprocessed, cleaned, tokenize, and standardized. TF-IDF is used to capture the rare, unique terms and weight them according to their relative frequency across the dataset. LDA is applied to discover latent topics in the dataset, categorizing recipes based on ingredient patterns. Cosine similarity calculates the similarity between user input and recipe ingredients, considering both direct match and topic-based relationships. Finally, the top N recipes with the highest relevance scores are selected and recommended to the user.

4.1. Data Sources

The system uses two primary sources of data. The publicly available Food.com recipe dataset from Kaggle.com, containing information about recipes such as name, ingredients, tags, steps, and descriptions. Manually collected dataset about 400 traditional Myanmar recipes with detail ingredients list and instructions were added to the dataset.

4.2. Data Preprocessing

To prepare the data for modeling, preprocessing focus on the ingredient's column. Raw ingredient string often contains quantities (e.g., 1cup, 2tsp, 2g) and common cooking terms such as fresh, pan, cut etc. These were removed to improve the quality of feature extraction. Words are reduced to their base form by lemmatization, lowercased to maintain the consistency, and tokenized. Then the preprocessed ingredients list from all recipes will form the corpus. The corpus was then used for TF-IDF vectorization and LDA topic model. Since the system takes the input as a list of ingredients (e.g., *chicken, garlic, onion*), word-level tokenization is applied.

Table 1. User Input and Top Recommended Output Sample

User Input Ingredients	Top 5 Recommended Recipes
chicken, garlic, onion	arroz caldo, spicy chicken nuggets, chicken fried rice, week night chicken, san francisco chicken
tomato, cheese, basil	eggplant baked with tomato, sun dried tomato and basil feta spread, lasagna with zucchini noodles, phyllo pizza Sonoma, basil bread slice pizzas
garlic, ginger, chicken, noodle	eggs a la king, tuna lasagna, Shan Noodle, the perfect ginger garlic paste subru uncle has taught me, cheat n eat vietnamese chicken soup
beef, carrot, potato	lobscouse lobscows, baltimore hash, greatest chips french fries on earth, very easy hash brown, ultimate butternut squash soup
fish, lemon, dill	golden topped fish fillets, old fashioned fish cakes with sour cream sauce, lemon dill mayonnaise sauce, lemon herb and fish risotto, citrus roasted fish with capers
egg, flour, sugar	german butter s cookies, plain muffins, feather pancakes, sugar bread, qwik coffee cake

pork, soy sauce, ginger	a pork marinade, grilled korean pork chops omac, oriental cabbage rolls, slow cooker teriyaki ribs, polynesian spareribs crock pot
shrimp, garlic, chili	mexicali shrimp kebabs, gingered shrimp with corn broccoli, szechwan shrimp chili shrimp, shrimp teriyaki, green onion rice
spinach, feta, olive oil	mediterranean spinach salad, athenian spinach, super spinach side, spinach mash with garlic and fetta, spinach cucumber feta and red onion salad
rice, bean, corn	iraqi white bean stew, beans beans and more beans, veggie chili, secret ingredient beef veggie chili, vanilla latte

As shown in Table 1, the system can suggest various cuisine and dishes according to the input. It is capable of recommending the recipes that exactly match the user input and also suggest recipes with different topics, such as bake, soup, salad_ showcasing the system's ability to capture broader culinary themes.

4.3. Evaluation on Top 5 Recommended Recipes

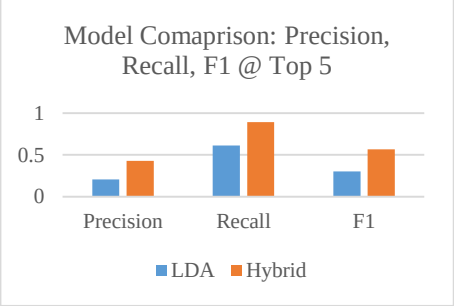


Figure 3. Model Comparison of the Two Model (LDA and Hybrid (LDA+TF-IDF approach))

Figure 3 presents the performance comparison between the two model based on top 5 recommended recipe. According to the results, the hybrid approach of LDA and TF-IDF works better than the pure LDA approach.

The precision, recall, and f1 scores are calculated for both approaches to test the performance.

4.3.1. Precision

Precision measures how many of the recommended recipes are actually relevant to the user. Precision is calculated as (4):

$$\text{Precision@K} = \frac{Relevant\ items\ in\ top\ K\ recommendations}{K} \quad (4)$$

4.3.2 Recall

Recall measures how many relevant recipes the system managed to retrieve from all relevant recipes. Recall is calculated as (5):

$$\text{Recall@K} = \frac{Relevant\ items\ in\ top\ K\ recommendation}{Total\ number\ of\ relevatn\ items} \quad (5)$$

4.3.3. F1-score

Recall measures how many relevant recipes found are accurate by combining both precision and recall. F1-score is calculated as (6):

$$\text{F1@K} = 2 \times \frac{Precision@K \times Recall@K}{Precision@K + Recall@K} \quad (6)$$

4.4. Evaluation with Topic Visualization (LDA)

The topic coherence scores are calculated to evaluate the quality and interpretability of topics generated by the LDA model. By computing coherence scores across different number of topics, the optimal number of latent topics can be selected, helping to avoid both underfitting (too few broad topics) and overfitting (too many narrow or irrelevant topics).

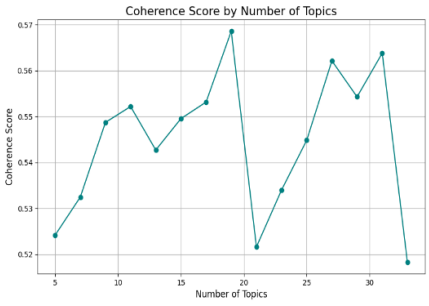


Figure 4. Coherence Scores

The coherence scores greater than 0.5 means that the topics have moderate quality. The scores above 0.6 are considered good topics and above 0.7 are very coherent and interpretable topics. Figure 4 show the coherence scores for the LDA model, number of topics near 0.57 coherence score is selected as best model for LDA.

Table 2. Coherence Scores for Each Topic

Number of Topics	Coherence Scores
5	0.524174583
7	0.53246033

9	0.54872371
11	0.552190477
13	0.542711988
15	0.549551561
17	0.553159975
19	0.568603972
21	0.521612369
23	0.534018166
25	0.544856336
27	0.562140936
29	0.554293777
31	0.563814415
33	0.518294331

Table 2 presents the coherence scores computed for various topic counts in the LDA topic modeling process. Higher coherence scores indicate that the top words in each topic tend to co-occur more frequently and semantically align, making the topics more meaningful. Therefore, 19 topics were selected as the optimal value for topic modeling.

Table 3. Top Ingredients Generated by Topic Model

Topic	***Top Ingredients***
Topic 1	pepper, onion, ground, beef, bean
Topic 2	cheese, chicken, parmesan, garlic, sauce
Topic 3	bread, yogurt, tamarind, cauliflower, floret
Topic 4	cheese, cream, butter, flour, milk
Topic 5	juice, lemon, orange, sugar, water

Table 3 shows that the LDA model can capture the certain culinary themes (e.g., “Italian cuisine, Desserts, Juice, Main dishes), reflecting the model’s ability to cluster semantically similar recipes.

5. Conclusion

This paper presented the content-based recipe recommendation system that integrates traditional TF-IDF vectorization with LDA topic modeling to recommend recipes based on user-provided ingredients. Combining LDA with TF-IDF enhances the recipe recommendation by capturing both ingredients significance and latent ingredient relationships. The evaluation results show that the hybrid model has higher performance over the pure LDA approach. The coherence score analysis also show that the model is capable of capturing meaningful and interpretable culinary themes within the dataset. The system delivers relevant, interpretable suggestions without relying on user history.

However, the system can only take the clean ingredients as input and there is no user profiling, limiting the system to provide personalized recommendations according to users’ historical data. Future work may include user profiling integrating user preferences, nutritional constraints and past interaction data for more personalized and health aware recipe recommendations. It may involve extending the hybrid model with collaborative filtering or deep learning architectures (e.g., Word2Vec or BERT) to capture deeper semantic relationships among ingredients.

References

[1] J. Freyne and S. Berkovsky, “Recommending Food: Reasoning on Recipes and Ingredients,” User Modeling, Adaptation, and Personalization, pp. 381–386, 2010.

[2] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, “Recipe recommendation using ingredient networks,” Proceedings of the 3rd Annual ACM Web Science Conference on - WebSci ’12, 2012

[3] M. Ge, M. Elahi, I. Fernaández-Tobías, F. Ricci, and D. Massimo, “Using Tags and Latent Factors in a Food Recommender System,” Proceedings of the 5th International Conference on Digital Health 2015, May 2015

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” Journal of Machine Learning Research, vol. 3, no. Jan, pp. 993–1022, 2025, Accessed: May 28, 2025.

[5] S. Chhipa, V. Berwal, T. Hirapure, and S. Banerjee, “Recipe Recommendation System Using TF-IDF,” ITM Web of Conferences, vol. 44, p. 02006, 2022

[6] M. Kāle and E. Agbozo, “Utility of Large-Scale Recipe Data in Food Computing,” Baltic Journal of Modern Computing, vol. 9, no. 2, 2021.

[7] J. N. Bondevik, K. E. Bennin, Ö. Babur, and C. Ersch, “A systematic review on food recommender systems,” Expert Systems with Applications, vol. 238, p. 122166, Mar. 2024.

[8] J. Idakwo, Joshua Babatunde Agbogun, and Taiwo Kolajo, “A Survey on Recommendation System Techniques,” UMYU Scientifica, vol. 2, no. 2, pp. 112–119, Jun. 2023.

[9] M.A.Khan, E.Rushe, B.Smyth, D.Coyle, “Personalized, Health-Aware Recipe Recommendation: An Ensemble Topic Modeling Based Approach”, ACM July 2019

A Corpus-Based Method for Revealing Category Overlap In Research Paper

Kyawt Kay Thwe Htoo
University of Computer Studies (Taunggyi)
Taunggyi, Myanmar
kyawtkaythwehtoo@ucstgi.edu.mm

Hsu Mon Kyi
Faculty of Computer Science
Polytechnic University (Maubin)
hsumonkyi@ucsmub.edu.mm

Abstract

Computer Scientific papers need to be organized by researchers so relevant information can be found without any trouble. Organizing these papers into categories is not easy because some fields share many similar terms. This study focuses on building a special collection of computer science research papers which are meant to be categorized into five types: Artificial Intelligence, Image Processing, Networking and Cybersecurity, Software Engineering, and Distributed Systems. By building these datasets, it becomes easy to realize how topics and words overlap between categories and thus complicate automatic classification. The paper describes the process of collecting and preparation of the data and discusses examples of overlapping vocabulary which confuse the categories. The classification is done using TF-IDF for feature extraction and a machine learning model based on Support Vector Machines (SVM). However, understanding these overlaps helps explain why basic word-based methods find it hard to differentiate between categories cleanly. This work shows the advantage of constructing domain-specific corpora to analyze and tackle classification problems better in academic papers with which future research tools might be improved.

Keywords- Scientistic paper classification, Computer science corpus, Category overlap, TF-IDF, Support Vector Machine

1. Introduction

In today's digital age, the rapid growth of scientific publications—especially in computer science—has created a need for effective organization. Automatic classification helps group papers by field, making it easier for students, teachers, and researchers to access relevant information. [13] However, in fields like Artificial Intelligence, Image Processing, Software Engineering, Networking and Cybersecurity, and Distributed Systems, shared terminology often leads to misclassification. For example, terms like "neural network" or "object recognition" may appear in multiple fields, challenging basic text mining methods.

Many studies use general datasets, such as news articles or reviews, which differ from academic writing in format and vocabulary. [12] To address this, the proposed system builds a domain-specific dataset using titles and abstracts from actual computer science papers, with each labeled by its field for supervised learning.

The system uses TF-IDF to extract important terms and a Support Vector Machine (SVM) to classify papers into five categories. A key contribution is highlighting the difficulty caused by overlapping vocabulary, which can lead to errors. [6] This paper also demonstrates how to build a domain-specific corpus for classification.

The rest of the paper is organized as follows: Section 2 discusses classification challenges, Section 3 covers motivation and goals, Section 4 describes dataset creation, Section 5 presents the system workflow and overlapping issues, and Section 6 provides the conclusion and future work.

2. Related Works

Text mining and classification have been extensively studied, with various methods developed to improve accuracy. Kwangil Park (2020) proposed a deep learning model using word embeddings and CNNs, which requires large datasets and high computation. Amber Saxena et al. (2019) introduced a transformer-based multi-label classifier, but it lacks interpretability for lightweight systems. J. Rashid (2019) used IDF with fuzzy K-means for biomedical topic modeling, enhancing clustering but without explicit category labels..

Unlike many prior studies that rely on general-purpose corpora or deep learning models with high resource requirements, this study introduces a lightweight, interpretable, and

domain-specific system. By focusing on TF-IDF and SVM, this method achieves competitive accuracy while remaining computationally efficient. Additionally, it uniquely analyzes overlapping vocabulary between categories, offering insights often ignored in traditional approaches. This focus on overlap and interpretability sets this study apart.

3. Background Theory

This section outlines the key methods used in the system. Section 3.1 covers the challenges of classifying computer science papers due to overlapping vocabulary. Section 3.2 highlights the role of domain-specific corpora in improving accuracy. Section 3.3 provides an overview of the system. Section 3.4 explains how TF-IDF converts text into numbers and its limitations. Section 3.5 introduces the SVM model used for paper classification.

3.1. Challenges in Classifying Computer Science Papers with Overlapping Terms

In computer science, fields like AI, Image Processing, and Software Engineering often share technical terms, making automatic classification difficult. For instance, "neural network" appears in both AI and Image Processing papers. This overlap, visible in TF-IDF values (e.g., "model," "data," "learning"), reduces classification accuracy by confusing category boundaries [9]. Figure 1 shows shared top TF-IDF terms, confirming the issue. Building a domain-specific corpus helps address this, improving classification and relevance in search results.

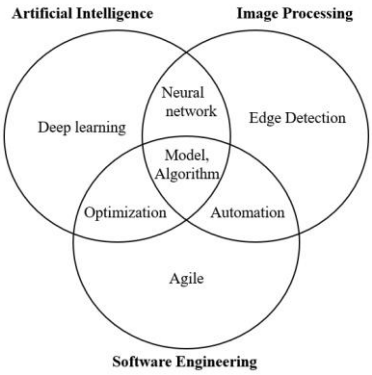


Figure 1. Example of Overlapping Terms

3.2. Importance of Domain-Specific Corpora in Improving Classification

General datasets like blogs or news differ from academic writing, leading to poor classification results. A domain-specific corpus using titles and abstracts from computer science papers helps the system learn relevant context and terminology. This improves TF-IDF feature extraction and SVM classification, even when categories share similar keywords. [2] Focusing on real research language avoids noise and enhances accuracy. [1][5]

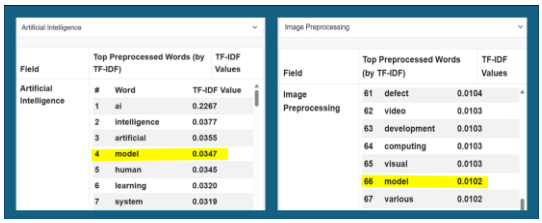


Figure 2. Vocabulary Overlap Between Research Categories

3.3. The Overview of the System

To sort computer science research papers into five main categories, this study focuses on building an automated system. It starts by collecting a large collection of papers, then doing the preprocessing method that will make the text easier to analyze. Next, words which have the highest weigh or being one of the most important in each paper are identified by using Term Frequency and Inverse Document Frequency (TF-IDF), which helps turn the text into vector forms. [6] After that, the system trains a machine learning model by using Support Vector Machines which is to learn how to classify the papers correctly. [9] Finally, the system's accuracy is tested, showing that it can effectively organize lots of research papers and help users find information faster.

The preprocessing step includes several key operations: (1) Lowercasing – converting all text to lowercase to ensure uniformity; (2) Punctuation and Symbol Removal – eliminating special characters and symbols that do not contribute to meaning; (3) Stop Word Removal – discarding common words such as 'and,' 'the,' and 'of' that carry little semantic value; and (4) Stemming or Lemmatization – reducing words to their root or base forms to unify variations. These processes standardize the input text, allowing the system to focus on meaningful terms and

improving the accuracy of feature extraction and classification.

Once the text has been preprocessed, the system moves on to extracting key features using the TF-IDF (Term Frequency–Inverse Document Frequency) method. [6] This technique helps identify the most relevant terms in each paper by measuring how important a word is within a document and across the entire collection.

After feature extraction, a classification model is applied to categorize the papers. This study uses a Support Vector Machine (SVM), a commonly used machine learning method for text classification tasks.

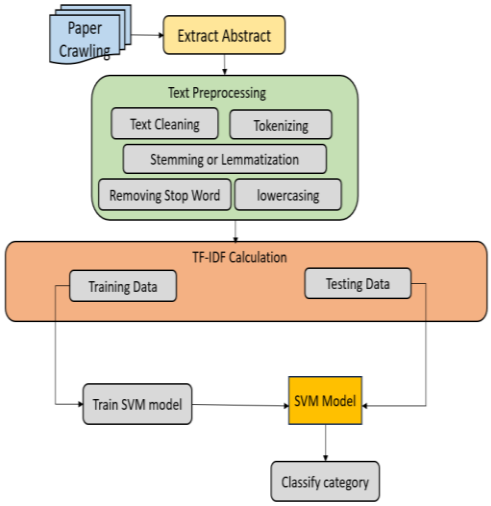


Figure 3. System Flow Diagram

3.4. Feature Representation

Research papers in computer science often contain complex terminology and overlapping vocabulary across subfields. To classify such unstructured text, it must first be converted into structured numerical features. This study uses Term Frequency–Inverse Document Frequency (TF-IDF), a common method in text mining, for that purpose.

TF-IDF measures a word's importance by combining its frequency in a document with its rarity across the dataset. It is particularly useful in distinguishing computer science papers that share common terms like "data" or "algorithm." For example, it highlights field-specific terms such as "convolutional neural network" for image processing or "load balancing" for distributed systems, making classification more precise. [10]

The TF-IDF output is a high-dimensional, sparse vector representing each document, where each dimension corresponds to a term from the vocabulary. These vectors are then fed into the classification model. [4] This transformation process, called vectorization, bridges the gap between text and machine learning.

Despite limitations, TF-IDF is a simple, interpretable, and efficient method for feature extraction. It forms the foundation of this system, allowing classifiers like Support Vector Machine (SVM) to detect patterns and accurately categorize academic texts. [14]

$$\mathrm{TF} = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}} \quad (1)$$

$$\mathrm{IDF} = \log\left(\frac{\text{number ofdocuments in corpus}}{\text{number of documents in the corpus contain the term}}\right) \quad (2)$$

$$\mathrm{TF-IDF} = TF * IDF \quad (3)$$

Table 1. TF-IDF Calculation

Term	Total TF-IDF Document 1	Total TF-IDF Document 2	Total TF-IDF Document 3
artificial	0.0151	0	0
intelligence	0.0151	0	0
evolving	0.0136	0	0
data	0.0100	0.0185	0
finance	0.0136	0	0
cybersecurit	0	0.0251	0
software	0	0	0.0265
engineering	0	0	0.0265

3.5. Classification of Research Papers

Following the conversion of unstructured text from computer science research papers into structured numerical data using TF-IDF, the next step is to apply a machine learning algorithm that can learn from these patterns to classify each paper into a predefined category. [10] This study uses the Support Vector Machine (SVM) for classification due to its strong performance, generalization ability, and capacity to handle high-dimensional feature spaces. [9]

There are several key components in the system, from preprocessing to classification. SVM is a supervised learning algorithm widely used in text classification. Its goal is to find an optimal hyperplane that separates data points into categories in a high-dimensional space. [9] Documents are mapped based on TF-IDF values. SVM then identifies the best decision boundary to separate categories. [4] In high dimensions, this boundary becomes a flat plane. The

algorithm maximizes the margin between support vectors—data points closest to the boundary. A wider margin generally leads to better classification. [14]

This study focuses on categorizing papers into five fields: Artificial Intelligence, Image Processing, Networking and Cybersecurity, Software Engineering, and Distributed Systems. A key challenge is vocabulary overlap. Terms like "model," "data," or "algorithm" appear across categories, making separation based on frequency alone difficult. SVM handles this by detecting patterns in word usage and capturing subtle differences. [3]

A manually compiled dataset of research articles is used. [11] SVM learns from TF-IDF features to distinguish categories.

The training set includes 1500 articles, with 300 per category. For training and testing, the dataset was split using an 80:20 ratio. Thus, each category included 240 articles for training and 60 articles for testing, maintaining equal class distribution. Each article is vectorized using TF-IDF and labeled for training. SVM learns to classify papers by detecting contextual patterns, even when shared terms like "learning" or "network" appear across fields such as AI and Cybersecurity. Performance is evaluated using accuracy, precision, recall, and F1-score, and the trained model is used to classify new papers after preprocessing and vectorization. [3]

SVM is effective with sparse, high-dimensional data and handles overlapping vocabulary well, supporting accurate research paper classification. [9] In experiments, a linear kernel was used with C = 1.0, selected via grid search. For comparison, KNN (K = 5, cosine similarity) underperformed due to high dimensionality, and Multinomial Naïve Bayes (α = 1.0) showed limited accuracy due to its independence assumption.

4. Data Statistics and Analysis

Research papers were collected from publicly available computer science repositories such as IEEE Xplore, ACM Digital Library, arXiv, and university databases. Only English-language papers published between 2018 and 2024 were considered to ensure topical relevance. Titles and abstracts were extracted and manually labeled into five categories based on the journal's indexing and keyword analysis.

The dataset used in this study contains a total of 1500 research papers, equally divided among five categories: Artificial Intelligence (300), Image Processing (300), Networking and Cybersecurity (300), Software Engineering (300), and Distributed Systems (300).

The average number of words per abstract is approximately 185, while titles average around 12 words. After preprocessing, the final vocabulary contained approximately 8,200 unique terms.

Analysis of TF-IDF weights revealed a substantial overlap in commonly used terms. For instance, the word "model" appeared in over 70% of documents across all categories. Additionally, some fields had higher lexical similarity, Artificial Intelligence and Image Processing shared over 40% of their top 50 weighted terms.

These statistics underscore the challenge of cleanly separating categories based solely on keyword frequency and reinforce the need for contextual classification.

The dataset was split into an 80:20 ratio for training and testing. Each category contributed 240 papers for training and 60 for testing.

5. Implementation and Testing New Paper

The developed system for computer science research paper classification is implemented with a user-friendly interface to demonstrate its functionality. This allows users to easily input the title and abstract of a new paper and receive a predicted category.

To test the system's performance on new, unseen documents, a web-based interface was created. As shown in Fig. 4, this interface allows a user to input the "Title" and "Abstract" of a research paper. The user can also select the desired SVM Kernel, with "Linear" being the kernel used in this study.

Upon clicking the "Categorize Paper" button, the system processes the input text. The title and abstract undergo the same preprocessing steps as the training data. Subsequently, the TF-IDF vector for the new document is computed based on the vocabulary learned from the domain-specific corpus. This feature vector is then fed into the trained Linear SVM model.

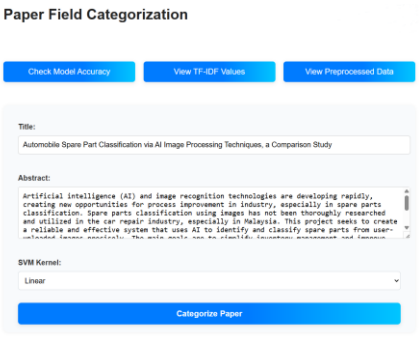


Figure 4. User Interface for Categorization

The output of the classification process for a test case, as depicted in Fig. 5, illustrates the system's prediction. For a paper titled "Automobile Spare Part Classification via AI Image Processing Techniques, a Comparison Study," the system predicted two primary fields with associated probabilities: Image processing (61%) and Artificial Intelligence (39%).

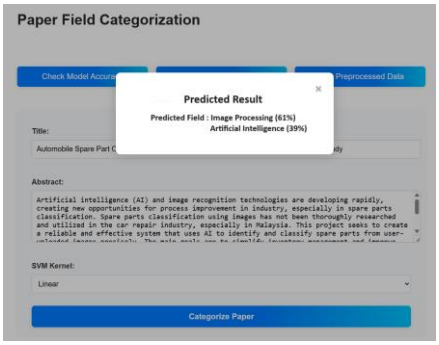


Figure 5. Predicted Result for New Document Categorization

Table 2 shows the precision, recall, F1-score, and accuracy for each research category, demonstrating the SVM model's strong and consistent performance across all five fields.

Table 2. Classification Performance

Category	Precision	Recall	F1-score	Accuracy
Artificial Intelligence	0.88	0.84	0.86	80%
Image Processing	0.89	0.86	0.87	82%
Networking and	0.95	0.94	0.94	94%
Software Engineering	0.87	0.85	0.86	83%
Distributed Systems	0.85	0.82	0.83	79%
Overall Average	0.89	0.86	0.87	85.6%

To further support the selection of SVM, this study also included a comparative evaluation with other common machine learning classifiers such as Naïve Bayes and K-Nearest Neighbors (KNN). While all models were trained using the same TF-IDF feature vectors derived from titles and abstracts, the performance results showed a significant difference. The SVM classifier achieved an average accuracy between 88% and 91%, outperforming Naïve Bayes which achieved 75% to 78%, and KNN which ranged from 70% to 73%. Moreover, SVM yielded higher precision and recall scores, indicating better consistency in identifying the correct category. These results demonstrate that while simpler models struggle with the overlapping vocabulary and high-dimensional data found in scientific papers, SVM remains robust and reliable. This confirms that SVM is not only theoretically suitable but also empirically superior for the task of categorizing domain-specific academic documents.

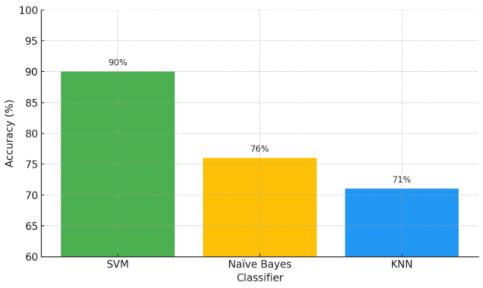


Figure 6. Accuracy Comparison

Classification accuracy across five fields was tested with datasets of 500, 1000, and 1500 papers. As shown in Figure 6, accuracy improved with dataset size. Artificial Intelligence rose from 60% to 80%, while Networking and Cybersecurity reached 94% accuracy with 1500 papers. Software Engineering and Distributed Systems had lower accuracy, likely due to overlapping technical terms affecting category separation.

The implementation clearly visualizes the classification process and shows how the SVM model with TF-IDF features classifies new research papers, even when documents span multiple related fields.

Table 3. Accuracy Comparison Across Fields with Varying Numbers of Papers

Field	500 papers	1000 papers	1500 papers
Artificial intelligence	60%	73%	80%
Image Processing	61%	72%	82%
Networking and	68%	80%	94%
Software Engineering	61%	70%	83%
Distributed Systems	59%	69%	79%

6. Conclusions

This study presents a corpus-based approach to categorize computer science research papers while addressing the challenge of category overlap. By preprocessing text and transforming it using TF-IDF, the system captures term importance across papers. A Support Vector Machine classifier then distinguishes among five research domains, despite shared vocabulary. The model reveals subtle patterns in language that indicate category blending, offering insight into overlapping thematic content. The results demonstrate effective classification performance and show the feasibility of using machine learning to uncover ambiguities in research boundaries. Future improvements may incorporate semantic techniques to enhance overlap detection and classification accuracy further.

References

[1] Al-Habib H.; Imah E., Riskyana Dewi Intan P., "*Text Processing Using Support Vector Machine for Scientific Research Paper Content Classification*", Advances in Intelligent Systems Research, Proceedings of the 1st International Conference on Neural Networks and Machine Learning (ICONNSMAL 2022), May 2023

[2] An Yang; Li Sujian*;" SciDTB: Discours Dependency TreeBank for Scientific Abstracts*", Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, July 2018

[3] Cahyani, D. E.; Patasik, I.; "*Text Vectorization in Sentiment Analysis: A Comparative Study of TF-IDF and Word2Vec from Amazon Fine Food Reviews.*", *ITM Web of Conferences*, *70*, 03001, 2025

[4] Che Mohd Safawi; N. U., & Shafie, N. A.; " *Performance of TF-IDF for Text Classification Reviews on Google Play Store*", Shopee. Journal of Computing Research and Innovation, 9(2), 13–22, 2024Sannella, M. J. 1994 *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington

[5] **Chen, Z.; Wang, Z.; Lin, Z.; "***Comparing ELM with SVM in the field of sentiment classification of social media text data*", Research Collection School Of Computing and Information Systems, Conference Proceeding Article, 2018Brown, L. D., Hua, H., and Gao, C. 2003. *A widget framework for augmented interaction in SCAPE*

[6] Das, M.;Selvakumar, K.; Alphonse, P. J. A; "*A Comparative Study on TF-IDF Feature Weighting Method and Its Analysis Using Unstructured Dataset*", 5th International Conference on Computational Linguistics and Intelligent Systems, 2023

[7] Fan H.; QinY., "*Proceedings of the 2018 International Conference on Network, Communication, Computer Engineering*", Advances in Intelligent Systems Research, May 2018

[8] Garcia A.; Jose Manuel G.; " *Not just about size - A Study on the Role of Distributed Word Representations in the Analysis of Scientific Publications*", arXiv:1804.01772, 2018

[9] Gomez, J.; Alfaro, C.; Ortega, F.; "*Adapting Support Vector Optimisation Algorithms to Textual Gender Classification.*". TOP, 32, 463–488, 2024

[10] Liang, M.; Niu, T; "*Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs*" Procedia Computer Science, 208, 460–470, 2022

[11] Pradana, A. W.;Hayaty, M.; " *The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts.*", Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(4), 375–380, November 2019

[12] Sadat, M.; Caragea, C.; "*Hierarchical Multi-Label Classification of Scientific Documents*", Computation and Language(cs.CL)(2022), https://doi.org/10.48550 /arXiv. 2211.02810

[13] Zhang, F.; Wu, S; "*An Instance-based Plus Ensemble Learning Method for Classification of Scientific Papers*", https://doi.org/10.48550/arXiv.2409.14237, 2024

[14] Zhou, H.; "*Research of Text Classification Based on TF-IDF and CNN-LSTM.*", Journal of Physics: Conference Series, 2171(1), 012021, 2022