e-ISSN: 2583-3472

Enhanced Support Vector Machine for Spam Email Classification

Patrick L. Marcos [1*], Dana Justine D. Pacatang [2]

[1, 2] Student, Computer Science Department, College of Information Systems and Technology, University of the City of Manila, Manila, Philippines
*Corresponding Author Email: plmarcos2020@plm.edu.ph

Abstract

Support Vector Machines (SVM) have shown strong performance in various classification tasks. However, SVM's performance deteriorates when faced with high-dimensional data due to the curse of dimensionality, where the increasing number of features reduces the model's ability to generalize and increases computational complexity. This study addressed this challenge by using an enhanced SVM model that incorporates a Term Frequency-Inverse Document Frequency - Class Variance (TF-IDF-CV) feature extraction method applied in spam email classification. Unlike traditional methods, TF-IDF-CV considers class variance during feature extraction, which helps mitigate the negative effects of high-dimensional data. Experimental results demonstrate that the enhanced SVM outperforms traditional feature extraction techniques, including TF-IDF, Bag of Words, and Word2Vec, achieving an accuracy of 99.42%, precision of 99.43%, recall of 99.42%, and an F1-score of 99.42%. These results highlight the model's improved robustness and reliability, making it a promising solution for accurate and efficient spam detection in high-dimensional datasets.

Keywords

Classification, Curse of Dimensionality, Feature Extraction, High-Dimensional Data, Machine Learning, Support Vector Machine, TF-IDF.

INTRODUCTION

Support Vector Machine (SVM), developed by Vapnik, is a well-known algorithm widely utilized for classification due to its strong mathematical foundation rooted in risk minimization. Fundamentally, SVM identifies the ideal decision boundary or optimal hyperplane that separates data points into distinct categories. The best decision boundary is the one that maximizes the margin between these categories, with the nearest data points—called support vectors—playing a key role in defining the classifier [1]. In the context of document classification, SVMs are frequently employed to categorize documents into various predefined classes [2] and have proved to be more accurate than most other classification techniques [3].

While SVM is great for classifying high-dimensional data [4], it is still not immune to the curse of dimensionality (COD), as all machine learning algorithms suffer from it [5]. COD describes the difficulties that emerge when processing and structuring data in high-dimensional spaces [6]. In high-dimensional, low-sample-size contexts, SVM can lead to data piling which happens when the training data from each class gets projected onto the normal vector of the separating hyperplane in such a way that all points in a class end up at the same position along this vector [7][8].

Feature extraction is a key step in data preprocessing for SVM classifiers, helping to address COD and enhance model performance. It involves transforming high-dimensional datasets into more manageable forms by removing redundant and irrelevant features while preserving the most informative aspects of the original data [9]. For text data, techniques like Term Frequency-Inverse Document Frequency (TF-IDF) are commonly employed to convert raw text into numerical feature vectors, reflecting word significance based on its occurrence in a document compared to its frequency across the entire corpus [10]. However, traditional TF-IDF does not consider the class variance, which can limit its ability to effectively differentiate between categories in classification tasks.

To overcome this limitation, this study utilizes TF-IDF with Class Variance (TF-IDF-CV), an enhanced feature extraction method that integrates class-specific information into the feature weighting process. By incorporating class variance, TF-IDF-CV produces more representative features that capture distinctions between classes, improving SVM's ability to find the optimal decision boundary [11]. This enhancement not only helps reduce the impact of COD but also increases the overall accuracy and efficiency of SVM in text classification tasks, making it a more robust solution for high-dimensional data analysis.

LITERATURE REVIEW

Various studies have employed different algorithms for classification tasks. These studies provide a foundation for comparing improvements in SVM performance. While many studies focus on email classification, others explore broader applications of machine learning, such as language identification. This section reviews these studies, focusing on algorithm choices and the results obtained.

[12] examined the performance of various machine learning algorithms in identifying spam emails. Using a dataset of 962 emails for training and 260 emails for testing, the study evaluates six methods: k-nearest neighbors (k-NN), Decision Tree, Random Forest, Naïve Bayes, SVM, and Adaboost. The results indicate that SVM achieved the highest accuracy at 97.33%, with Random Forest closely behind at

97.30%. The study concludes that SVM and Random Forest are the most effective models for spam detection, with future work suggested for deep learning approaches and model fine-tuning.

[13] investigated the use of SVM in email spam detection, focusing on its ability to distinguish spam emails from non-spam ones. The proposed approach includes preprocessing techniques such as removing noise, numbers, and symbols, followed by feature extraction from email content and training an SVM classifier. The dataset used includes emails, though the exact size is not specified. The SVM model achieved 98% accuracy in classifying emails as spam or non-spam.

[14] examined the effectiveness of ML algorithms, specifically Random Forest and Support Vector Classification (SVC), a type of SVM, in classifying phishing emails. Using a dataset of 18,650 email samples obtained from Kaggle, the researchers experimented with different training-to-testing splits (60:40, 70:30, 80:20) and applied the TF-IDF Vectorizer to convert text into numerical representations. The findings revealed that SVC consistently outperformed Random Forest, achieving the highest accuracy of 97.52% with a 70:30 data split, compared to Random Forest's maximum accuracy of 96.57%. This study highlights the superior performance of SVC in phishing email classification, contributing valuable insights into enhancing cybersecurity measures.

[15] evaluated the performance of six machine learning algorithms—Decision Tree, Gaussian Naïve Bayes, k-NN, Logistic Regression, SVM, and Random Forest—for detecting hate speech on Twitter. Using a large dataset containing both positive and negative hate speech samples, the text data was processed using CountVectorizer before training and testing the models. The findings revealed that Random Forest, Decision Tree, and SVM outperformed other algorithms, achieving accuracies of 98.2%, 96.2%, and 95.5%, respectively. The study concludes that these models are effective at detecting hateful content on Twitter, with future work focusing on improving efficiency and expanding applicability to other social media platforms.

[16] explored language identification by evaluating different machine learning models and text vectorization methods. The study evaluated three models—Naïve Bayes, Logistic Regression, and SVM—along with two vectorization methods: BoW and TF-IDF. Using a dataset from Kaggle containing 10,367 samples in 17 languages, including four Indian languages, the researchers found that Logistic Regression with TF-IDF achieved the highest accuracy at 98.45%, followed by Naïve Bayes with BoW at 97.34%. The study concluded that Logistic Regression with TF-IDF performed best, but identified challenges with very short sentences and multilingual texts, suggesting areas for future work using larger corpora and LSTM methods.

In the context of feature extraction methods, [11] introduced the TF-IDF-CV approach, aiming to address limitations of traditional TF-IDF by refining how textual features are weighted. TF-IDF-CV incorporates class variance, which adjusts weights based on a word's distribution across categories and within documents. This approach emphasizes features that are uniquely characteristic of specific classes and evenly distributed in text, potentially enhancing classification accuracy. Although not widely adopted, TF-IDF-CV shows promise in boosting model performance, and this study applies it with SVM to improve classification accuracy.

[17] expanded on the limitations of statistical approaches like TF-IDF by leveraging semantic representation through Word2Vec for single-label (SLC) and multi-label (MLC) classification of research articles. The study utilized metadata features such as titles, keywords, and general terms from two datasets (JUCS: 1,460 articles; ACM: 86,116 articles) and introduced a semantic-based text representation method. Additionally, a data-driven approach was proposed to determine MLC thresholds without requiring domain expertise. The results showed that combining title and keywords achieved the highest accuracy for SLC (0.86 for JUCS, 0.84 for ACM) and MLC (0.81 for JUCS, 0.80 for ACM). The study concluded that semantic models improve classification performance compared to traditional statistical techniques while reducing cognitive dependence on expert-defined thresholds. However, computational inefficiency was identified as a limitation, suggesting the need for further optimization.

METHODOLOGY

This study aims to enhance spam detection by applying an optimized feature extraction technique with an SVM classifier. The methodology comprises four phases: preprocessing, feature extraction, model training, and evaluation. The effectiveness of the model is measured through essential indicators like accuracy, precision, recall, and F1-score to demonstrate the efficiency of the proposed approach.

Dataset:

This research uses a dataset originally prepared by V. Metsis, I. Androutsopoulos, and G. Paliouras [18], referred to here as the Metsis Dataset. It consists of 33,716 pre-labeled emails, with 16,545 classified as ham and 17,171 as spam, making the dataset balanced. The ham emails are exclusively derived from a subset of the Enron corpus, while the spam emails originate from four primary sources: the SpamAssassin corpus and the Honeypot project, contributing a combined total of 5,175 spam emails; Bruce Guenter's spam collection, adding 6,000 spam emails; and an additional 6,000 spam emails personally collected by G. Paliouras. In this dataset, each email combines both the subject line and body into a single text field, with ham assigned a label of 0 and spam a label of 1.

Preprocessing:

While the initial Metsis Dataset was preprocessed to a large extent, some inconsistencies remained, such as residual symbols, capital letters, and unnecessary whitespace. To address these problems, the researchers converted all text to lowercase, removed any extraneous symbols, and eliminated redundant whitespace. This additional preprocessing step ensured a uniform text format, facilitating consistent feature extraction and improving the robustness of the spam detection model [19].

Feature extraction:

Feature extraction converts raw email content into a structured form suitable for machine learning classifiers [20]. In this study, the researchers apply TF-IDF with class variance (TF-IDF-CV) as the primary feature extraction method. To evaluate its performance and effectiveness, TF-IDF-CV is compared against other widely used techniques, such as Word2Vec, Bag of Words (BoW), and the traditional TF-IDF approach [17]. These methods were selected as benchmarks to assess the improvements introduced by TF-IDF-CV with SVM.

Term Frequency Inverse Document Frequency (TF-IDF):

TF-IDF is a technique that sets weights to terms according to their relevance in a document relative to their frequency compared to their occurrence throughout the dataset. The core idea behind TF-IDF is that terms frequently appearing in a specific document but rarely in others are more informative and should be assigned a higher weight [21]. This technique is particularly effective for identifying distinguishing words in spam detection, as certain terms appear more commonly in spam emails than in non-spam (ham) emails.

The TF-IDF score for a term *i* within a document *j* is determined using the following formula:

$$TF-IDF_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i+1}\right) \tag{1}$$

Equation (1) [21], in which *N* represents the total count of documents in the corpus, and df_i refers to the frequency of term *i* across those documents. In this equation, the *TF* component measures a word's significance within a particular document, while the IDF_{ij} (inverse document frequency) component assesses its rarity across the corpus. By combining these two factors, TF-IDF emphasizes terms that are especially characteristic of individual documents, which enhances the model's effectiveness in differentiating between spam and ham emails.

Term Frequency Inverse Document Frequency with Class Variance (TF-IDF-CV)

The TF-IDF-CV weighting method builds upon the traditional TF-IDF by incorporating three distinct factors α, β, and γ that are designed to enhance the weighting by accounting for class-specific distribution, interclass relevance, and term variance, respectively [11]. These distribution factors collectively aim to improve the identification of distinguishing terms between classes, especially in cases of imbalanced datasets, such as spam and ham email classification [11][22][23].

For a given term *t* within a document *d*, the TF-IDF-CV weight assigned to *t* in *d* is expressed as:

$$\begin{aligned} TF-IDF-CV_{td} &= TF_{td} \times IDF_t \times \alpha_t \times \beta_t \\ &\times \gamma_t \end{aligned} \tag{2}$$

Each term in the formula has a specific role in emphasizing certain characteristics of words within the dataset. The α, β, and γ introduce class-level adjustments to the term weights based on distribution and variability among classes (spam and ham).

Category Distribution Factor (α)

The category distribution factor α_t accounts for the distribution of a term *t* across different classes, ensuring that terms disproportionately appearing in one class are given more weight. This factor is crucial for balancing the influence of classes with varying document counts, as it emphasizes terms prevalent in smaller classes, which are often underrepresented in traditional TF-IDF calculations [22][23]. While the current dataset is balanced, incorporating this factor ensures that the methodology remains effective for datasets with varying class distributions.

The α factor is computed as:

$$\alpha_t = \log_2\left(\frac{N}{n_{t,c}}\right) \tag{3}$$

where *N* represents the total count of documents, and $n_{t,c}$ denotes the count of documents in class *c* that contains term *t*. Category distribution is calculated for each class and aggregated across all classes to provide a comprehensive weight for terms based on class distribution [11].

Interclass Distribution Factor (β)

The interclass distribution factor β_t enhances the weighting by evaluating the presence of a term *t* across classes. This factor emphasizes terms that are specific to one class over others, as such terms are critical for distinguishing between classes in classification tasks [11][23].

The β factor is formulated as:

$$\beta_t = \log_2\left(2 + \frac{t_{2c}}{n_{2c}}\right) \tag{4}$$

where t_{2c} is the sum of term occurrences of *t* within class *c*, n_{2c} indicates how many documents in class *c* include term *t*. The interclass distribution considers the relative distribution of term *t* within each class and aggregates this distribution across all classes. By assigning higher weights to terms with concentrated occurrences in specific classes, β_t helps improve the model's effectiveness in distinguishing between classes.

Variance Distribution Factor (γ)

The variance distribution factor γ_t quantifies the dispersion of term t within a class, accounting for how the term's frequency varies across documents. Terms with higher variance are given more weight, as they are more likely to be central to distinguishing between classes, especially in high-dimensional spaces where such variance indicates a term's importance in capturing the class's defining characteristics [11][23].

The γ factor is calculated as:

$$\gamma_t = \log_2\left(\frac{\sum_{j=1}^{m}(t_j-\mu)^2}{n_3}\right) \quad (5)$$

where *m* denotes the positions of term *t* in documents, μ is the mean frequency of term *t* across the documents in a class, and n_3 refers to the total count of documents where *t* appears in that class. The variance distribution γ_t captures the variance of term distributions across all documents. By amplifying weights for terms with high variance, γ_t enhances the model's ability to recognize terms that exhibit unique patterns across documents within a class.

The final computation of TF-IDF-CV is represented as:

$$TF - IDF - CV_{td} = \frac{TF \times IDF \times \alpha_t \times \beta_t \times \gamma_t}{\sqrt{\sum_{i=1}^{|V|}(TF-IDF-CV_{td})^2}} \quad (6)$$

By integrating α, β, and γ, the researchers achieve a weighting scheme that dynamically adjusts term weights based on their relevance across different classes and the variance of their occurrence. In this formula the *V* is the vocabulary size. This normalization step uses the L2 normalization to ensure that each document's feature vector has a consistent length, which is critical for classification stability in high-dimensional spaces.

Model Training:

Once the features have been extracted through TF-IDF-CV, the next phase is training the machine learning model. The dataset is separated into training (80%) and testing (20%), ensuring that the model learns from the majority of the data while being evaluated on unseen samples. In this study, the researchers use linear SVM as the primary classifier. Additionally, Logistic Regression, Random Forest, and Naïve Bayes are included for comparison to assess classification performance and evaluate SVM's strengths and limitations against other widely used classifiers [15].

Support Vector Machine (SVM):

SVM falls under supervised machine learning algorithms and is primarily designed for classification tasks. SVM works by mapping the data points into a higher-dimensional space, enabling a hyperplane to distinguish between classes [1][3]. The model aims to maximize the margin between the closest data points of each class, known as the support vectors. These support vectors are the critical elements in defining the hyperplane, as they determine the margin and the classification decision boundary.

The hyperplane can be expressed as follows:

$$w \times x + b = 0 \quad (7)$$

where *w* represents the weight vector that determines the hyperplane's direction, *x* is the feature vector of the data instance, *b* is the bias term that repositions the hyperplane. The objective is to determine the best possible hyperplane that optimizes the separation margin between classes. This margin is the measured distance between the hyperplane and the immediate data points from each class, referred to as support vectors [21]. The optimal margin is given by:

$$M\arg in = \frac{1}{||w||} \quad (8)$$

In the case of linearly separable data, the optimization problem involves determining *w* and *b* that satisfy the constraints:

For data points of class 1 ($y_i = +1$):

$$w \times x + b \geq +1 \quad (9)$$

Any points or features belonging to class 1 should be on one side of the hyperplane with a margin of at least 1.

For data points of class -1 ($y_i = -1$):

$$w \cdot x + b \leq -1 \quad (10)$$

Any points or features belonging to class -1 are on the opposite side of the hyperplane with a margin of at least 1. The separation hyperplane is then chosen to maximize this margin while ensuring that the constraints are satisfied.

Evaluation:

During the model evaluation phase, classifier performance is measured using key metrics that assess effectiveness in distinguishing between spam and ham emails. These are drawn from the confusion matrix, which compares predicted labels with actual labels in the dataset [21]. The main evaluation criteria in this study include accuracy, precision, recall, and F1-score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$\Pr ecision = \frac{TP}{TP + FP} \quad (12)$$

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

$$F1 - score = 2 \times \frac{\Pr ecision \times Recall}{\Pr ecision + Recall} \quad (14)$$

These metrics are calculated from the four fundamental components of the confusion matrix: True Positives (TP) are the spam emails that the classifier correctly identifies as spam, while True Negatives (TN) refer to ham emails that are correctly identified as ham. False Positives (FP) occur when ham emails are mistakenly marked as spam, and False Negatives (FN) arise when spam emails are incorrectly labeled as ham. Together, these metrics provide a comprehensive assessment of the model's classification accuracy as well as its tendency to misclassify emails.

RESULTS AND DISCUSSION

This research trains the SVM model using TF-IDF-CV with the Metsis Dataset. The enhanced method demonstrated a high degree of effectiveness in classifying emails. As shown in Table 1, when comparing its performance to a standard TF-IDF feature extraction, BoW, and Word2vec, the TF-IDF-CV model achieved slightly superior results across key metrics, confirming its potential for improved email classification accuracy.

Table 1: Classification Performance Metrics for TF-IDF, BOW, Word2Vec, and TF-IDF-CV applied with SVM model

SVM Classification with Feature Engineering Algorithm					
Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Prediction**
Word2Vec	98.18	98.30	97.96	98.14	Benign
		98.06	98.84	98.21	Spam
BOW	98.21	98.48	97.85	98.17	Benign
		97.95	98.54	98.24	Spam
TF-IDF	98.86	99.48	98.19	98.83	Benign
		98.27	99.51	98.89	Spam
TF-IDF-CV	99.42	99.79	99.03	99.41	Benign
		99.08	99.80	99.43	Spam

The evaluation of feature extraction techniques—Word2Vec, BoW, traditional TF-IDF, and the enhanced TF-IDF-CV—revealed notable differences in classification accuracy and precision when applied to an SVM classifier for spam email detection. The BoW model attained 98.21% accuracy, with precision rates of 98.48% for benign and 97.95% for spam emails. Similarly, Word2Vec delivered an accuracy of 98.18%, with a precision of 98.30% for benign and 98.06% for spam emails. Traditional TF-IDF demonstrated further improvement, reaching 98.86% accuracy, with precision scores of 99.48% for benign and 98.27% for spam classifications. However, the enhanced TF-IDF-CV outperformed all other methods, achieving 99.42% accuracy, with precision rates of 99.79% for benign and 99.08% for spam emails. These results underscore the effectiveness of TF-IDF-CV in improving precision and accuracy, addressing high-dimensionality challenges in text classification, and enhancing SVM-based spam detection.

The enhanced TF-IDF-CV feature extraction technique was further applied to four machine learning classifiers—Logistic Regression, Multinomial Naïve Bayes, Random Forest, and SVM—was conducted to evaluate and identify the optimal model for spam email classification. The results, presented in Table 2, highlight notable variations in classifier performance across key metrics. These findings emphasize the influence of classifier selection when utilizing the TF-IDF-CV method for email classification.

Table 2: Classification Performance Metrics for Logistic Regression, Multinomial Naïve Bayes, Random Forest, and SVM Models applied with TF-IDF-CV

Models	**Classification Models with TF-IDF-CV**			
	Accuracy	**Precision**	**Recall**	**F1-Score**
Logistic Regression	98.89	98.92	98.87	98.89
Multinomial Naïve Bayes	99.04	99.03	99.04	99.04
Random Forest	98.86	98.86	98.86	98.86
Support Vector Machine	99.42	99.43	99.42	99.42

SVM demonstrated the highest performance metrics, achieving 99.42% accuracy, 99.43% precision, 99.42% recall, and 99.42% F1-score, indicating its strong capability to differentiate between spam and benign emails. This confirms SVM's compatibility with TF-IDF-CV, as it effectively leverages the extracted features for improved separation of classes. In comparison, Multinomial Naïve Bayes also performed well, reaching 99.04% accuracy, but with slightly lower precision and F1-scores. Logistic Regression and Random Forest achieved competitive accuracy rates of 98.86% and 98.89%, respectively; however, these models did not match the precision and F1-score of the SVM, suggesting they were less effective at capturing nuanced differences in the email data.

CONCLUSION

This study showed that TF-IDF-CV significantly improves SVM’s performance, achieving better accuracy, precision, recall, and F1-score compared to conventional approaches. The findings highlight TF-IDF-CV’s ability to address high-dimensional challenges, enhancing SVM's overall effectiveness. Future research could focus on refining TF-IDF-CV or exploring additional feature engineering techniques to further boost SVM's performance. This improvement in SVM's capabilities opens up possibilities for its application in other complex classification tasks.

ACKNOWLEDGEMENT

We thank Vivien A. Agustin, our thesis adviser, for her invaluable guidance and insights throughout the course of this research. Special thanks to Khatalyn E. Mata, our Thesis Writing subject professor, for her constructive feedback and support.

REFERENCES

[1] R. S. A. Corpuz, “Categorizing natural language-based customer satisfaction: An implementation method using Support Vector Machine and Long Short-Term Memory Neural Network,” *International Journal of Integrated Engineering*, vol. 13, no. 4, May 2021, doi: 10.30880/ijie.2021.13.04.007.

[2] R. Lupyani and J. Phiri, “Automated Document Classification for research HEI grant awards using Machine Learning”, *Zapuc Conference*, vol. 3, no. 1, pp. 90–95, Aug. 2023.

[3] H. T. Sueno, “Multi-class Document Classification using Support Vector Machine (SVM) Based on Improved Naïve Bayes Vectorization Technique,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3937–3944, Jun. 2020, doi: 10.30534/ijatcse/2020/216932020.

[4] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, “Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF,” *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, pp. 1–6, Oct. 2020, doi: 10.1109/icdabi51230.2020.9325685.

[5] M. E. Samadi, S. Kiefer, S. J. Fritsch, J. Bickenbach, and A. Schuppert, “A training strategy for hybrid models to break the curse of dimensionality,” *PLoS ONE*, vol. 17, no. 9, p. e0274569, Sep. 2022, doi: 10.1371/journal.pone.0274569.

[6] A. A. Awan, “The curse of dimensionality in machine learning: Challenges, impacts, and solutions,” *DataCamp*, Sep. 13, 2023. https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning (accessed Nov. 27, 2024).

[7] K. Egashira, K. Yata, and M. Aoshima, “Asymptotic properties of distance-weighted discrimination and its bias correction for high-dimension, low-sample-size data,” *Japanese Journal of Statistics and Data Science*, vol. 4, no. 2, pp. 821–840, Aug. 2021, doi: 10.1007/s42081-021-00135-x.

[8] L. Shen, M. J. Er, and Q. Yin, “Classification for high-dimension low-sample size data,” *Pattern Recognition*, vol. 130, p. 108828, Jun. 2022, doi: 10.1016/j.patcog.2022.108828.

[9] S. Ayesha, M. K. Hanif, and R. Talib, “Overview and comparative study of dimensionality reduction techniques for high dimensional data,” *Information Fusion*, vol. 59, pp. 44–58, Jan. 2020, doi: 10.1016/j.inffus.2020.01.005.

[10] Y. M. M. Seethalakshmi, S. Andavar, and R. S. P. Raj, “A survey on feature extraction techniques, classification methods and applications of sentiment analysis,” *Brazilian Archives of Biology and Technology*, vol. 66, Jan. 2023, doi: 10.1590/1678-4324-2023220654.

[11] X. Zhang, Y. Shi, and H. Wei, “Research on TFIDF algorithm based on weighting of distribution factors,” *Journal of Physics Conference Series*, vol. 1621, no. 1, p. 012007, Aug. 2020, doi: 10.1088/1742-6596/1621/1/012007.

[12] S. I. Manzoor and J. Singla, “A comparative analysis of machine learning techniques for spam detection,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 3, pp. 810–814, Jun. 2019, doi: 10.30534/ijatcse/2019/73832019.

[13] S. Pandey, A. Taralekar, R. Yadav, S. Deshmukh, and S. Suryavanshi, “E-mail spam detection and classification using SVM,” International Journal of Computer Science and Information Technologies, vol. 11, no. 1, pp. 6–8, 2020, [Online]. Available: https://www.ijcsit.com/docs/Volume%2011/vol11issue01/ijcsit2020110102.pdf

[14] C. Umam, L. B. Handoko, and F. O. Isinkaye, “Performance analysis of Support Vector Classification and Random Forest in phishing email classification,” *Scientific Journal of Informatics*, vol. 11, no. 2, pp. 367–374, May 2024, doi: https://doi.org/10.15294/sji.v11i2.3301.

[15] S. Das, K. Bhattacharyya, and S. Sarkar, “Performance Analysis of Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest and SVM on Hate Speech Detection from Twitter,” International Research Journal of Innovations in Engineering and Technology, vol. 07, no. 03, pp. 07–03, Jan. 2023, doi: 10.47001/irjiet/2023.703004.

[16] A. Bhansali, A. Chandravadiya, B. Y. Panchal, M. H. Bohara, and A. Ganatra, “Language identification using combination of machine learning algorithms and vectorization techniques,” *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 1329–1334, Apr. 2022, doi: 10.1109/icacite53722.2022.9823628.

[17] G. Mustafa, M. Usman, L. Yu, M. T. Afzal, M. Sulaiman, and A. Shahid, “Multi-label classification of research articles using Word2Vec and identification of similarity threshold,” Scientific Reports, vol. 11, no. 1, Nov. 2021, doi: 10.1038/s41598-021-01460-7.

[18] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam Filtering with Naive Bayes - Which Naive Bayes?,” *Conference on Email and Anti-Spam*, Jan. 2006. Available: http://cs.txstate.edu/~v_m137/docs/papers/ceas2006_paper_corrected.pdf

[19] P. Peace, J. Chris, and L. Victor, “Data preprocessing for AI models,” *ResearchGate*, Nov. 2024. https://www.researchgate.net/publication/385707249_Data_Preprocessing_for_AI_Models

[20] A. Khalid, M. Hanif, A. Hameed, Z. A. Smiee, M. M. Alnfiai, and S. M. M. Alnefaie, “LogiTriBlend: A Novel Hybrid Stacking Approach for Enhanced Phishing Email Detection using ML Models and Vectorization Approach,” *IEEE Access*, p. 1, Jan. 2024, doi: 10.1109/access.2024.3518923.

[21] N. Sutriawan, N. Muljono, N. Khairunnisa, Z. Alamin, T. A. Lorosae, and S. Ramadhan, “Improving performance sentiment movie review classification using hybrid feature TFIDF, N-Gram, information gain and support Vector Machine,” *Mathematical Modelling and Engineering Problems*, vol. 11, no. 2, pp. 375–384, Feb. 2024, doi: 10.18280/mmep.110209.

[22] W. Dai, “Classification and analysis of literary works based on distribution weighted term frequency-inverse document frequency,” *Journal of Physics Conference Series*, vol. 1941, no. 1, p. 012018, Jun. 2021, doi: 10.1088/1742-6596/1941/1/012018.

[23] L. He, Y. Long, N. Wang, Z. Ma, and Z. Ma, “Research on Network Media news visualization based on FDCD-TFIDF weighting Algorithm,” *Journal of New Media and Economics.*, vol. 1, no. 1, pp. 100–109, Jan. 2024, doi: 10.62517/jnme.202410114.