Paradigma, Vol. 28, No. 1, March 2026, P-ISSN 1410-5063, E-ISSN: 2579-3500, Page. 42-49
Published: LPPM Universitas Bina Sarana Informatika

PARADIGMA

website: http://jurnal.bsi.ac.id/index.php/paradigma/

Comparative Analysis of Email Spam Detection Using SVM with TF-IDF and Word2Vec on Multilingual Datasets

Kaifa Ahlal Katamsyi[1], Ahmad Taufiq Akbar[1], Andi Nurkholis[1], Hari Prapcoyo[1], Bagus Muhammad Akbar[1], Shoffan Saifullah[2]

[1]Department of Informatics, Universitas Pembangunan Nasional Veteran Yogyakarta, Yogyakarta, Indonesia
[2]Faculty of Computer Science, AGH University of Krakow, Krakow, Poland

ARTICLE INFORMATION

Artikel History:
Received: 10-02-2026
Revised: 03-03-2026
Accepted: 23-03-2026
Available Online: 31-03-2026

Keyword:

Hyperparameter Optimization
Spam Email Classification
Support Vector Machine
TF-IDF
Word2Vec

ABSTRACT

The rapid growth of email communication has increased the prevalence of spam emails, which can disrupt productivity and compromise information security. This study presents a comparative analysis of two text representation methods—TF-IDF and Word2Vec—for spam email classification using a Support Vector Machine (SVM) with a Radial Basis Function kernel. The experiments utilized Indonesian and English email datasets totaling 5,421 emails, split into 75% training and 25% testing sets. Two scenarios were evaluated: baseline with default parameters and after hyperparameter optimization using Grid Search combined with K-Fold Cross Validation. The results indicate that TF-IDF consistently outperformed Word2Vec across both languages, achieving the highest accuracy of 0.9562 on the English dataset after tuning. Word2Vec showed substantial improvement following parameter adjustment, reducing the performance gap with TF-IDF. The findings highlight the importance of hyperparameter optimization for enhancing the quality of feature representations and improving classification performance. This study also demonstrates that TF-IDF provides more stable results across different linguistic contexts, while Word2Vec benefits significantly from careful tuning. The results provide practical insights for implementing efficient spam email detection systems in multilingual environments. Future research could explore additional classifiers, deep learning approaches, and contextual embeddings to further improve classification accuracy and robustness.

Corresponding Author:

Ahmad Taufiq Akbar,
Department of Informatics,
Universitas Pembangunan Nasional Veteran Yogyakarta,
Padjajaran (Ring Road Utara) Street No. 104, Sleman, Special Region of Yogyakarta, Indonesia, 55283,
Email: ahmad.taufiq@upnyk.ac.id

INTRODUCTION

The progression of information technology has profoundly altered human communication, notably with the advent of email as a rapid and effective digital communication tool. Email is widely used across various contexts, including personal, business, and organizational communication (Thomas, Chen, & Iacobucci, 2022) that demands speed and reliability in information delivery. However, as email utilisation expands, the issue of spam has arisen, potentially hindering user productivity, straining system resources, and presenting significant information security risks (Sari & Sutabri, 2023). In fact, 45.6% of worldwide emails in 2023 were classified as spam. (Mahusin & Prilliadi, 2025). Spam emails typically contain irrelevant advertisements, fraudulent attempts, and even malicious content (Tusher, Ismail, Rahman, Alenezi, & Uddin, 2024) that can harm users if not detected accurately and promptly.

Various text classification methods have been employed to identify spam emails automatically, including machine learning algorithms such as Naïve

DOI: https://doi.org/10.31294/p.v28i1.12339

Bayes, Logistic Regression, and Support Vector Machine. Among these algorithms, SVM is recognised for its exceptional efficacy in managing text data, owing to its capacity to establish an ideal decision border between spam and non-spam categories (Tohari, Harini, Yaqin, Santoso, & Crysdian, 2023). SVM is highly favored for spam detection because it efficiently manages the complex, high-dimensional matrices produced by natural language processing (NLP) techniques without compromising computational speed or accuracy. In machine learning for text classification, text representation methods are essential for transforming textual input into numerical formats suitable for algorithmic processing (Gasparetto, Marcuzzo, Zangari, & Albarelli, 2022). A prevalent representation method is Term Frequency-Inverse Document Frequency (TF-IDF), which quantifies text based on word occurrence frequency and the significance of words inside a document (Chamira, 2022). Although effective in capturing statistical patterns of keywords, TF-IDF possesses intrinsic limitations as it fails to capture semantic links between words and concentrates exclusively on word presence, disregarding the contextual framework in which words occur (Jaiswal & Das, 2021).

To address these limitations, word embedding-based approaches such as Word2Vec can be utilized as an alternative text representation. Word2Vec is a technique that encodes words as numerical vectors, effectively capturing semantic links and contextual associations based on their co-occurrence patterns inside sentences (S. Styawati et al., 2022). This approach consists of two primary architectures: Continuous Bag of Words (CBOW) predicts a word based on its contextual environment, while Skip-Gram forecasts the context from a specified target word. With its ability to capture semantic relationships, Word2Vec is expected to provide a richer text representation for spam email classification tasks.

Most prior studies evaluated TF-IDF and Word2Vec separately or within a single language setting, limiting generalizability across multilingual contexts (Arifiandy & Fahmi, 2024). Additionally, many studies did not systematically apply hyperparameter optimization to the SVM classifier, leaving the full potential of these feature representations unexplored. This study addresses these limitations by comparing TF-IDF and Word2Vec under optimized SVM parameters on both Indonesian and English email datasets, aiming to determine which method consistently achieves superior performance across languages.

This work intends to provide a comparative examination of TF-IDF and Word2Vec text representation approaches for spam email categorization utilizing the Support Vector Machine algorithm. The evaluation is performed on email datasets in both Indonesian and English, under baseline conditions as well as after parameter optimization using the Grid Search method. The main goal is to find the text representation method that performs best in accuracy, precision, recall, and F1-score, and to examine how parameter optimisation affects TF-IDF and Word2Vec in multilingual spam email classification.

RESEARCH METHOD

This study utilizes a quantitative research technique that commences with data collection, followed by preprocessing, feature extraction via TF-IDF and Word2Vec, classification using SVM, and model evaluation, as seen in Figure 1.

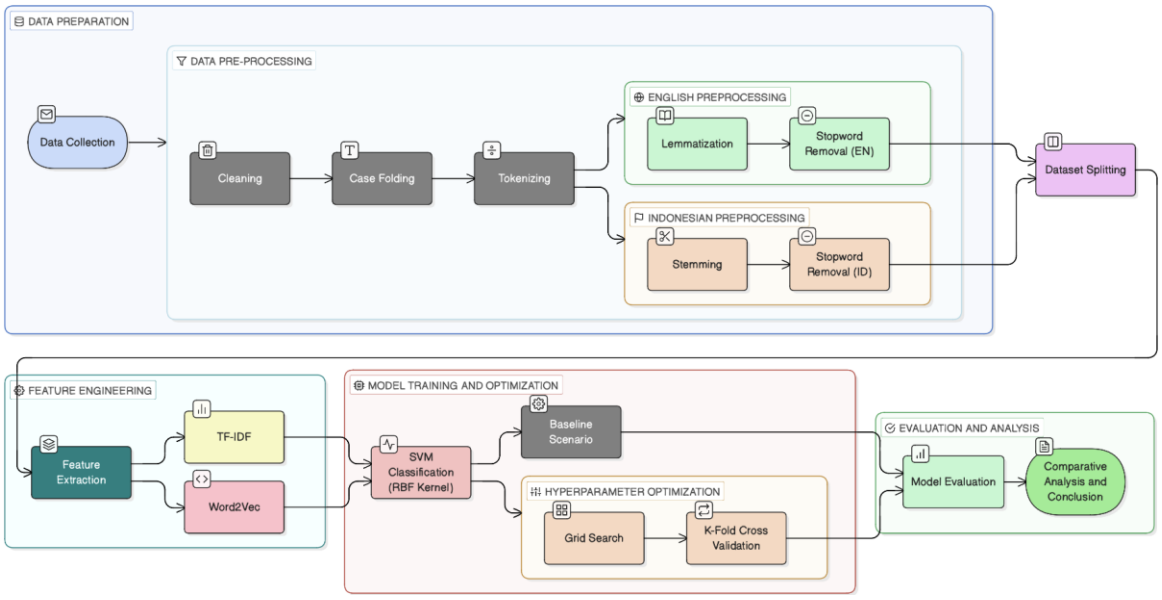

Figure 1. Research Stages

1. Data Collection

This study employs data consisting of Indonesian and English emails classified into two categories: spam and ham. The English dataset was

obtained from the public SpamAssassin dataset on Kaggle and was subsequently translated into Indonesian to create an Indonesian-language variant. In addition, primary data were collected through web scraping from the researcher's email account and through manual collection of publicly available raw email examples. This primary data collection yielded 623 emails, which were then merged with the secondary dataset. In total, the data collection process resulted in 5,421 emails, comprising 3,627 ham emails and 1,794 spam emails. Combining multiple data sources was intended to produce a dataset that is more representative of real-world conditions. Examples of the data used in this study are presented in Table 1.

Table 1. Example of dataset

No	Email	Class
1	Good morning, just wanted to let you know that the project deadline has been extended to next Friday. Let me know if you need any help.	0
2	Urgent! Your bank account has been accessed from an unknown location. Please click here to secure your account immediately.	1
3	Dear team, please find attached the minutes of our last meeting for review. Let me know if you have any questions.	0

To enhance data quality and credibility, all translated emails were reviewed for semantic accuracy, and duplicates were removed. For the scraped data, manual verification was conducted to ensure correct labeling of spam and ham instances. Stratified sampling was applied to maintain the original class distribution during dataset splitting. Despite these efforts, potential biases remain due to translation artifacts and the limited representativeness of scraped emails, which may not fully reflect real-world linguistic or thematic variations. These limitations were considered in the analysis and discussed in the interpretation of results.

2. Data Pre-processing

This stage is designed to enhance data quality and accuracy prior to subsequent processing (Maharana, Mondal, & Nemade, 2022). This stage involves pre-processing operations including as cleaning, case folding, tokenization, lemmatization, stemming, and stopword elimination.

a. Cleaning

The cleaning process aims to remove irrelevant elements from the text, such as numbers, symbols, and URLs that do not support the analysis (Tan, Lee, & Lim, 2023). This step helps reduce noise in the data and ensures that only meaningful information is processed in subsequent stages.

b. Case Folding

Case folding converts all letters to lowercase to ensure consistency and to avoid duplication caused by different capitalization (Andi Nurkholis, Alita, & Munandar, 2022). For example, the words “Email” and “email” will be treated as the same after this process.

c. Tokenizing

Tokenizing is the step of breaking text into its smallest units, such as words or tokens, using separators such as spaces (Andi Nurkholis et al., 2022). Each resulting token represents a distinct part of the text that will be analyzed in subsequent steps.

d. Lemmatization

Lemmatization is applied to the English dataset to convert words to their correct base form or lemma (Pant, Sharma, & Kundu, 2024), such as converting “running” to “run.” This process takes word context within a sentence into account to produce a more accurate base form, which is particularly useful in English due to its many inflected forms.

e. Stemming

Stemming is applied to the Indonesian dataset to reduce words to their base form by removing affixes (Isnain, Sulistiani, Hurohman, Nurkholis, & Styawati, 2022). For example, “berlari” is reduced to “lari,” and “membangun” to “bangun.” This process simplifies the analysis by reducing word variation caused by affixation, which is common in Indonesian.

f. Stopword Removal

Stopword removal removes conjunctions and pronouns that have no relevance for the analysis (Isnain et al., 2022). This process is performed after tokenizing by comparing each token against a predefined stopword list.

3. Dataset Splitting

In the dataset splitting stage, the data were divided into two subsets, namely training and testing sets, using a 75%–25% proportion (75% for training and 25% for testing). This split was performed using a stratified split to preserve the class distribution of spam and ham in both subsets, thereby reducing the risk of biased evaluation. Details of the training and testing set sizes, along with the class distributions for the Indonesian and English datasets, are presented in Table 2.

Table 2. Dataset splitting

Dataset	Training	Testing	Class dist. in Training	Class dist. in Testing
Indonesian	4065	1356	2715 (Spam), 1351 (Ham)	907 (Spam), 449 (Ham)
English	4065	1356	2715 (Spam), 1351 (Ham)	907 (Spam), 449 (Ham)

4. Text Representation Using TF-IDF

TF-IDF method is a strategy employed to assess the relevance of a term to a document by providing a weight to each word (Maulidya Prastita Syah, Ajeng Puspa Wardani, Mohammad Idhom, & Trimono, 2025). TF-IDF integrates two principles: the frequency of a word's occurrence in a text and the

inverse frequency of documents that include that word. Term Frequency $tf(w,d)$ is seen to possess proportional significance relative to the total frequency of its occurrences inside a text or document. IDF is a token weighting technique that quantifies the dispersion of a token within a corpus of texts. The TF-IDF method represents text in a tabular format, with each feature corresponding to an individual word (Chamira, 2022). TF measures the occurrence frequency of a word in relation to the total word count in a document, whereas IDF determines the logarithm of the total document count divided by the number of documents containing the target word (t).

$$idf_t = log\,\frac{N}{df_i} \quad (1)$$

$$tf - idf(t,d) = tf \times idf \quad (2)$$

N marks the total count of documents, df_t indicates the document frequency value, and $tf(t,d)$ reflects the term frequency of term t in document d.

5. Text Representation Using Word2Vec

Word2Vec is a text representation method that transforms words into numerical vectors (embeddings) (Abubakar & Umar, 2022). Its primary objective is to construct word representations such that words frequently appearing in similar contexts will have vectors that are closer together in the vector space. Word2Vec has two main architectures: CBOW and Skip-Gram (Choudhary & Beniwal, 2021) as seen in Figure 2. CBOW forecasts a target word utilizing its contextual surroundings by leveraging the distribution of neighboring words, whereas Skip-Gram attempts to predict context words from a given target word. Word2Vec initiates the process by converting words into input vectors through one-hot encoding. These input vectors are then projected onto a hidden layer using a weight matrix. Subsequently, the model computes the output values and converts them into probabilities using the softmax function. The resulting probabilities are compared against the target to calculate the error, and the weights are then updated through a learning process known as backpropagation.

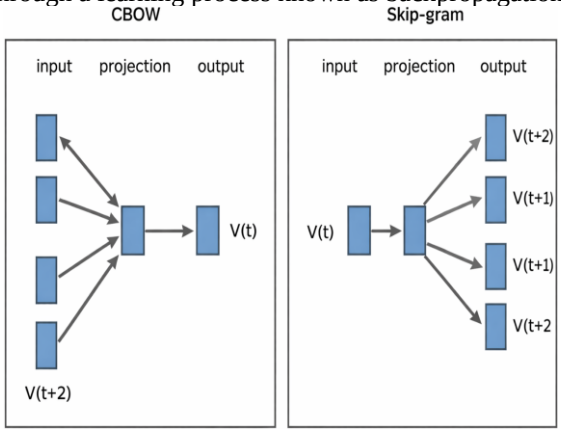

Figure 2. Arsitektur Word2Vec

a. Continuous Bag of Words

CBOW uses multiple context words as input to form a representation in the hidden layer (Feng, Hu, Kamigaito, Takamura, & Okumura, 2022). At this stage, the context vectors are first summed and then projected using the weights from the input to the hidden layer and averaged. The equation for constructing the hidden layer in CBOW is expressed as follows:

$$h = \frac{1}{C} W^T (x_1 + x_2 + \cdots + x_c) \quad (3)$$

Where h is hidden layer vector, W is weight matrix, and x_c is converted vector of the target word. Once the value of h is obtained, the next step is to compute the output and convert it into probabilities using the softmax function to generate the target word prediction. Subsequently, the error between the predicted output and the actual target word is calculated to update the weights.

b. Skip-Gram

Skip-Gram processes a single word as input and then predicts the surrounding context words (Xia, 2023). In the initial stage, the input word is converted into an input vector x, which is then projected onto the hidden layer using the weights from input to hidden layer. The hidden layer equation is expressed as follows:

$$h = W^T \times x \quad (4)$$

Where, h represents hidden neuron, x is input vector, and W^T is transpose of the weight matrix from the input layer to the hidden layer. Subsequently, the hidden layer value is multiplied by the weight matrix connecting the hidden layer to the output layer to get the output score for the j-th row:

$$u_j = W'^T \times h \quad (5)$$

Where, u_j is output at the j-th row, h is hidden neuron, and W'^T is transpose of the weight matrix from the hidden layer to the output layer. The resulting output is then transformed using the softmax function to obtain probabilities:

$$y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})} \quad (6)$$

Where, y_j is softmax output at the j-th row, u_j is output at the j-th row, $u_{j'}$ is output across all rows, and V is total number of unique words (vocabulary size). Next, the difference between the output and the target is used to calculate the error. The error calculation can be expressed as follows:

$$e = \sum_1^c (y_j - x_c) \quad (7)$$

Where, e is error value, x_c is vector value of the context word, and y_j is softmax output at the j-th row. The error value is used in the weight update process (backpropagation), enabling the model to learn increasingly refined word embeddings that better capture contextual patterns in the textual data.

6. SVM Classification

Introduced by Vapnik in 1992, SVM functions to classify both linear and non-linear data by finding a discriminant function in the form of a hyperplane that separates two classes based on the distance between data vectors (A. Nurkholis et al., 2022). The vectors used to construct this hyperplane are called support vectors, which enable SVM to generalize well to new data (Andi Nurkholis, Styawati,

Alim, Saputra, & Ferriyan, 2025). SVM is based on the principle of Structural Risk Minimisation (SRM) and is designed to determine the optimal hyperplane that separates two classes within the input space (Rokhman, Berlilana, & Arsi, 2021). In the prediction process, SVM assigns labels to data according to the relevant class. The efficacy of SVM is significantly contingent upon the parameters employed, such as the C value, epsilon, and kernel type (Andi Nurkholis, Styawati, & Suhartanto, 2024). Several kernels available in SVM provide flexibility in selecting an appropriate decision boundary function.

In this study, the Radial Basis Function (RBF) kernel was employed for SVM. The selection of the RBF kernel was based on the characteristics of the text features produced by Word2Vec embedding and TF-IDF, which are generally non-linear in nature. The RBF kernel is also considered more stable as it requires only two main hyperparameters (C and γ). Furthermore, a previous study by (Rizky, Jondri, & Lhaksmana, 2023) demonstrated that the RBF kernel yields superior performance compared to other kernels in text-based classification tasks. The model training was carried out using the Sequential Minimal Optimization (SMO) algorithm built into scikit-learn, with the feature representation consisting of averaged Word2Vec vectors that had undergone standardization.

7. Evaluation

A comparison study was performed utilising the measures of accuracy, precision, recall, and F1-score obtained from the confusion matrix. The baseline findings were compared with those obtained following parameter optimisation to identify the most effective and stable text representation approach for spam email categorisation. Evaluation criteria were utilised to examine the model's classification performance from various dimensions, including prediction accuracy, the ability to identify the positive class, and the balance between precision and recall. After obtaining the True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP) values from the confusion matrix, the subsequent step is to calculate the evaluation metrics (Azhari, Situmorang, & Rosnelly, 2021). The calculations using the formulas for accuracy, precision, and recall are presented as follows.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$Precission = \frac{TP}{TP+FP} \quad (10)$$

$$F1 - score = \frac{(2*Precission*recall)}{(precission+recall)} \quad (11)$$

RESULTS AND DISCUSSION

1. Parameter Optimization

Parameter optimization was executed via the Grid Search technique in conjunction with K-Fold Cross Validation. This study optimised the parameters of TF-IDF and Word2Vec, which serve as feature extraction techniques. The ideal parameters for TF-IDF include ngram_range, min_df, max_df, and max_features; for Word2Vec, they consist of vector_size, window, min_count, and sg. The parameter value ranges for each method are displayed in Tables 3 and 4.

Table 3. TF-IDF Hyperparameter Value Ranges

Hyperparameter	Type	Value
ngram_range	*Tuple*	(1,1), (1,2), (1,3)
min_df	*Int*	2, 5, 10, 15
max_df	*Float*	0.6, 0.75, 0.9, 1.0
max_features	*Int*	1000,2000,3000,4000

Table 4. Word2Vec Hyperparameter Value Ranges

Hypermarameter	Type	Value
vector_size	*Int*	100,200,300,400,500
window	*Int*	2, 5, 10, 15, 25
min_count	*Int*	2, 5, 10, 15, 25
sg	*Int*	1, 0

2. Model Evaluation Results

The evaluation was conducted under two scenarios: baseline (default parameters without optimization) and after hyperparameter optimization (tuning). The evaluation was applied to two dataset groups: Indonesian and English. A summary of the evaluation results for each dataset and scenario is presented in Table 5.

Under the baseline scenario (default parameters), SVM performance with TF-IDF and Word2Vec representations showed a fairly clear difference across both datasets as shown in Figure 3. On the Indonesian dataset, the SVM+TF-IDF

Table 5. Model Evaluation

Dataset	**Representation**	**Configuration**	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Indonesian	TF-IDF	Baseline	0.9340713	0.9440806	0.9075128	0.9225401
		After Tuning	0.95030750	0.94653321	0.94088057	0.94356234
	Word2Vec	Baseline	0.8814268	0.8732454	0.8550199	0.8628628
		After Tuning	0.94268142	0.93234602	0.94025598	0.93587154
English	TF-IDF	Baseline	0.9419434	0.9521635	0.9175295	0.9319446
		AfterTuning	0.95621156	0.95581778	0.94472856	0.94993359
	Word2Vec	Baseline	0.8929889	0.8867009	0.8681698	0.8763127
		After Tuning	0.94534041	0.93423906	0.94453702	0.93902089

combination attained an Accuracy of 0.9341, Precision of 0.9441, Recall of 0.9075, and F1-Score of 0.9225. Conversely, the SVM+Word2Vec combination yielded inferior metrics, achieving an Accuracy of 0.8814, Precision of 0.8732, Recall of 0.8550, and F1-Score of 0.8629. A same trend was noted in the English dataset. In the baseline setup, the SVM+TF-IDF combination attained an Accuracy of 0.9419, Precision of 0.9522, Recall of 0.9175, and F1-Score of 0.9319, whereas the SVM+Word2Vec combination produced an Accuracy of 0.8930, Precision of 0.8867, Recall of 0.8682, and F1-Score of 0.8763.

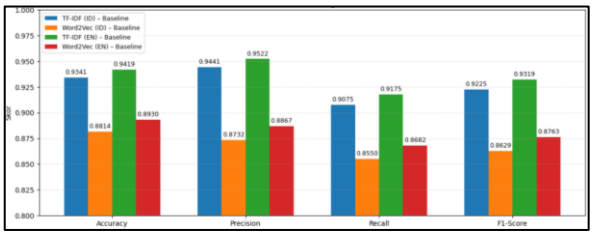

Figure 3. Model Performance Comparison with Baseline Parameters

The observed differences can be explained by the nature of the feature representations. TF-IDF relies on word frequency and document importance, which tends to capture discriminative patterns even in multilingual datasets. In contrast, Word2Vec embeddings are highly dependent on training quality and parameter settings, making them less stable under default configurations. These findings align with previous studies, which reported that TF-IDF generally provides more consistent results in text classification tasks, whereas Word2Vec benefits significantly from parameter tuning (Arifiandy & Fahmi, 2024; Styawati Styawati et al., 2022).

Overall, the baseline results indicate that the choice of feature representation can influence the quality of class separation by SVM. TF-IDF constructs features based on word weights that highlight important words within a document, allowing the model to capture relatively strong discriminative patterns from the initial configuration. In contrast, Word2Vec produces embeddings whose quality is highly dependent on training configuration (vector size, min count, sg). When the parameters remain at their default values, the resulting semantic representation is not necessarily optimal for supporting the classification process on the given data.

After hyperparameter tuning, the performance of both feature representations improved across all metrics on both the Indonesian and English datasets as shown in Figure 4. On the Indonesian dataset, the SVM+TF-IDF combination improved from an Accuracy of 0.9341 to 0.9503 and an F1-Score of 0.9225 to 0.9436. An improvement was also observed in Recall, which increased from 0.9075 to 0.9409, indicating an enhanced ability of the model to correctly identify instances belonging to the target class.

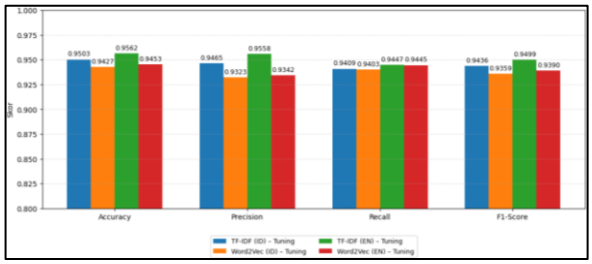

Figure 4. Model Performance Comparison with Parameter Optimization

These results indicate that hyperparameter tuning substantially enhances the quality of feature representations, allowing Word2Vec to approach TF-IDF performance. The implications are twofold: first, for practitioners implementing multilingual spam detection systems, careful parameter optimization is critical to achieving high accuracy, especially when using embeddings. Second, these findings suggest that TF-IDF remains a robust baseline for multilingual datasets, while Word2Vec can be advantageous if sufficient tuning resources are available. The study extends prior research by demonstrating the importance of systematic optimization across languages, highlighting the practical relevance of these methods for real-world email classification systems.

On the same dataset, the SVM+Word2Vec combination experienced a substantial improvement after tuning. Accuracy increased from 0.8814 to 0.9427, while the F1-Score rose from 0.8629 to 0.9359. Additionally, Recall improved from 0.8550 to 0.9403. This indicates that after reconfiguring the Word2Vec parameters, the model became significantly better at identifying instances that should belong to a particular class. On the English dataset, improvements were also observed for both methods. The SVM+TF-IDF combination improved from an Accuracy of 0.9419 to 0.9562 and an F1-Score of 0.9319 to 0.9499. Meanwhile, the SVM+Word2Vec combination improved from an Accuracy of 0.8930 to 0.9453 and an F1-Score of 0.8763 to 0.9390, with Recall increasing from 0.8682 to 0.9445.

Accordingly, hyperparameter tuning has been demonstrated to enhance the quality of the feature representations employed, thereby directly impacting SVM classification performance. These improvements indicate that appropriate parameter configuration can help feature representations capture textual data characteristics more effectively, both in the Indonesian and English language contexts.

CONCLUSION

This study provides a comprehensive evaluation of TF-IDF and Word2Vec for spam email classification using Support Vector Machine across Indonesian and English datasets. The analysis demonstrates that hyperparameter tuning not only improves overall model performance but also reduces the performance gap between Word2Vec and TF-IDF, highlighting the critical role of parameter optimization

in embedding-based text representations. TF-IDF consistently delivers stable and robust results across languages, while Word2Vec exhibits substantial improvement when optimized, indicating its potential in capturing semantic relationships once training parameters are carefully adjusted. The scientific contribution of this research lies in systematically comparing these two feature representation methods under optimized conditions across multilingual datasets, providing empirical evidence of their relative strengths and limitations. Practically, these findings inform the design of effective spam detection systems, guiding practitioners in selecting appropriate text representations and emphasizing the importance of tuning for real-world deployment. For future research, investigations could extend to additional classifiers and deep learning architectures, such as BERT, FastText, or contextual embeddings, to capture richer semantic features. Evaluating these methods on more diverse and imbalanced multilingual datasets would further validate their robustness and generalizability. Moreover, cross-lingual adaptation strategies could be explored to enhance model applicability in real-world multilingual email environments.

REFERENCES

Abubakar, H. D., & Umar, M. (2022). Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec. *SLU Journal of Science and Technology*, *4*(1 & 2), 27–33. https://doi.org/10.56471/slujst.v4i.266

Arifiandy, R., & Fahmi, H. (2024). Perbandingan Vektorisasi Deteksi Spam Email Menggunakan Bag of Word, TF IDF, dan Word2Vec pada Multinomial Naïve Bayes. In *Syntax: Jurnal Informatika* (Vol. 13). https://doi.org/10.35706/syji.v13i01

Azhari, M., Situmorang, Z., & Rosnelly, R. (2021). Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *5*(2), 640. https://doi.org/10.30865/mib.v5i2.2937

Chamira, S. (2022). Implementasi Metode Text Mining Frequency-Invers Document Frequency (Tf-Idf) Untuk Monitoring Diskusi Online. *Journal of Informatics, Electrical and Electronics Engineering*, *1*(3), 97–102. https://doi.org/10.47065/jieee.v1i3.353

Choudhary, K., & Beniwal, R. (2021). Xplore Word Embedding Using CBOW Model and Skip-Gram Model. *2021 7th International Conference on Signal Processing and Communication (ICSC)*, 267–270. IEEE. https://doi.org/10.1109/ICSC53193.2021.9673321

Feng, Y., Hu, C., Kamigaito, H., Takamura, H., & Okumura, M. (2022). A Simple and Effective Usage of Word Clusters for CBOW Model. *Journal of Natural Language Processing*, *29*(3), 785–806. https://doi.org/10.5715/jnlp.29.785

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A Survey on Text Classification Algorithms: From Text to Predictions. *Information*, *13*(2), 83. https://doi.org/10.3390/info13020083

Isnain, A. R., Sulistiani, H., Hurohman, B. M., Nurkholis, A., & Styawati, S. (2022). Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *8*(2), 299. https://doi.org/10.26418/jp.v8i2.54704

Jaiswal, M., Das, S., & Khushboo, K. (2021). Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(4), 3168. https://doi.org/10.11591/ijece.v11i4.pp3168-3175

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91–99. https://doi.org/10.1016/j.gltp.2022.04.020

Mahusin, M., & Prilliadi, H. (2025). *Harmonising ASEAN's Anti-spam Regulations: Strategies for Effective Cross-border Enforcement and Enhanced Regional Cooperation*. Retrieved from https://www.eria.org/research/harmonising-asean-s-anti-spam-regulations--strategies-for-effective-cross-border-enforcement-and-enhanced-regional-cooperation

Maulidya Prastita Syah, Ajeng Puspa Wardani, Mohammad Idhom, & Trimono. (2025). Perbandingan Representasi Teks Tf-Idf Dan Bert Terhadap Akurasi Cosine Similarity Dalam Penilaian Otomatis Jawaban Berbasis Teks. *Data Sciences Indonesia (DSI)*, *5*(1), 47–59. https://doi.org/10.47709/dsi.v5i1.6021

Nurkholis, A., Styawati, Sitanggang, I. S., Jupriyadi, Matin, A., & Maulana, P. (2022). SVM Multi-Class Algorithm for Soybean Land Suitability Evaluation. *2022 International Conference on Information Technology Research and Innovation, ICITRI 2022*. https://doi.org/10.1109/ICITRI56423.2022.9970216

Nurkholis, Andi, Alita, D., & Munandar, A. (2022). Comparison of Kernel Support Vector Machine Multi-Class in PPKM Sentiment Analysis on

Twitter. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *6*(2), 227–233. https://doi.org/10.29207/resti.v6i2.3906

Nurkholis, Andi, Styawati, S., Alim, S., Saputra, H., & Ferriyan, A. (2025). Sentiment Analysis of COVID-19 Booster Vaccines on Twitter Using Multi-Class Support Vector Machine. *Applied Information System and Management (AISM)*, *8*(1), 29–36. https://doi.org/10.15408/aism.v8i1.42911

Nurkholis, Andi, Styawati, S., & Suhartanto, A. (2024). Firefly Algorithm for SVM Multi-class Optimization on Soybean Land Suitability Analysis. *JOIV : International Journal on Informatics Visualization*, *8*(2), 592. https://doi.org/10.62527/joiv.8.2.1860

Pant, V. K., Sharma, R., & Kundu, S. (2024). An overview of Stemming and Lemmatization Techniques. In *Advances in Networks, Intelligence and Computing* (pp. 308–321). London: CRC Press. https://doi.org/10.1201/9781003430421-31

Rizky, F. M., Jondri, J., & Lhaksmana, K. M. (2023). Twitter Sentiment Analysis of Kanjuruhan Disaster using Word2Vec and Support Vector Machine. *Building of Informatics, Technology and Science (BITS)*, *5*(1), 219–227. https://doi.org/10.47065/bits.v5i1.3612

Rokhman, K. A., Berlilana, B., & Arsi, P. (2021). Perbandingan Metode Support Vector Machine Dan Decision Tree Untuk Analisis Sentimen Review Komentar Pada Aplikasi Transportasi Online. *Journal of Information System Management (JOISM)*, *3*(1), 1–7. https://doi.org/10.24076/JOISM.2021v3i1.341

Sari, P., & Sutabri, T. (2023). Analisis kejahatan online phising pada institusi pemerintah/pendidik sehari-hari. *Jurnal Digital Teknologi Informasi*, *6*(1), 29. https://doi.org/10.32502/digital.v6i1.5620

Styawati, S., Nurkholis, A., Aldino, A. A., Samsugi, S., Suryati, E., & Cahyono, R. P. (2022). Sentiment Analysis on Online Transportation Reviews Using Word2Vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm. *2021 International Seminar on Machine Learning, Optimization, and Data Science, ISMODE 2021*. https://doi.org/10.1109/ISMODE53584.2022.9742906

Styawati, Styawati, Nurkholis, A., Winarko, E., Rahmanto, Y., Reza, M. A., & Ismail, I. (2022). Sentiment Analysis of Indonesian Government Policy using Support Vector Machine-Word2Vec. *2022 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*. Jakarta: Institute of Electrical and Electronics Engineers (IEEE).

Tan, K. L., Lee, C. P., & Lim, K. M. (2023). A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. *Applied Sciences*, *13*(7), 4550. https://doi.org/10.3390/app13074550

Thomas, J. S., Chen, C., & Iacobucci, D. (2022). Email Marketing as a Tool for Strategic Persuasion. *Journal of Interactive Marketing*, *57*(3), 377–392. https://doi.org/10.1177/10949968221095552

Tohari, H., Harini, S., Yaqin, M. A., Santoso, I. B., & Crysdian, C. (2023). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Produktivitas Padi. *Journal of Computer System and Informatics (JoSYC)*, *5*(1), 175–183. https://doi.org/10.47065/josyc.v5i1.4538

Tusher, E. H., Ismail, M. A., Rahman, M. A., Alenezi, A. H., & Uddin, M. (2024). Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems. *IEEE Access*, *12*, 143627–143657. https://doi.org/10.1109/ACCESS.2024.3467996

Xia, H. (2023). Continuous-bag-of-words and Skip-gram for word vector training and text classification. *Journal of Physics: Conference Series*, *2634*(1), 012052. https://doi.org/10.1088/1742-6596/2634/1/012052