Optimized SMS Spam Detection Using SVM-DistilBERT and Voting Classifier: A Comparative Study on the Impact of Lemmatization

Sinar Nadhif Ilyasa, Alaa Omar Khadidos
Information Systems Department, King Abdul Aziz University, Jeddah, Kingdom of Saudi Arabia

***Abstract*—The rapid growth of digital communication has led to a surge in spam messages, particularly through Short Message Service (SMS). These unsolicited messages pose risks such as phishing and malware, necessitating robust detection mechanisms. This study focuses on a comparative analysis of machine learning models for SMS spam detection, with a particular emphasis on a proposed SVM-DistilBERT model enhanced by a voting classifier. Using the UCI SMS Spam dataset, the models are evaluated based on recall, accuracy, precision, and Receiver Operating Characteristic Area Under the Curve (ROC AUC) scores to assess their effectiveness in correctly identifying spam messages. By leveraging Optuna for hyperparameter optimization, the proposed model achieves superior performance, with an accuracy of 99.6%, surpassing traditional methods like SVM with TF-IDF Bi-gram and AdaBoost, which achieved 98.03%. The study also examines the effects of lemmatization and synonym data augmentation, with lemmatization shown to improve spam detection by reducing feature space redundancy and enhancing semantic understanding. To ensure transparency in decision-making, Local Interpretable Model-Agnostic Explanations (LIME) is applied. The results demonstrate that the optimized SVM-DistilBERT with the voting classifier offers a robust and effective solution for SMS spam filtering.**

***Keywords*—SMS spam detection; Support Vector Machine (SVM); DistilBERT; hyperparameter optimization; LIME**

I. Introduction

The advancement of digital communications in modern times has caused mass messaging, also known as spam, to become widespread. These messages flood inboxes across various channels, bringing severe security risks such as phishing and malware. The rise of spam is closely tied to technological advancements, with Short Message Service (SMS) emerging as one of the first mobile communication standards. As SMS usage grew, so did the prevalence of spam, creating an urgent need for effective spam detection methods.

A 2022 report [1] states that 68.4 million Americans, or 26% of the population, have been scammed via phone, compared to the previous year's 59.4 million (23%). Furthermore, 33% of people reported being involved in a phone scam, with about 20% falling for a con more than once. These scams not only have financial consequences but also affect productivity, mental health, and personal privacy. As mobile telecommunications have expanded, SMS spam has become a significant irritant, contributing to substantial losses in working time, network resource consumption, and performance costs [2].

The rise of spam undermines trust in mobile communication platforms and consumes valuable network resources and device storage. This highlights the necessity for effective spam reduction strategies to preserve user satisfaction and optimize resource utilization [3]. Implementing advanced spam detection mechanisms is crucial for protecting users and ensuring compliance with privacy regulations [4]. This underscores the importance of deploying sophisticated and explainable spam detection methodologies to bolster user trust and meet regulatory expectations.

Traditional spam detection methods have relied on rule-based systems [5], which offer limited success due to their inflexibility and inability to adapt to the evolving nature of spam tactics and content. This necessitates more sophisticated, adaptable, and accurate detection strategies. A survey of existing literature indicates ongoing efforts to combat this issue, yet it remains a significant challenge, highlighting the need for innovative approaches that can keep pace with the dynamic landscape of spam messaging.

Machine learning emerges as a promising solution to address the complex problem of spam detection [6]. These algorithms, by learning from categorized datasets, can effectively differentiate between spam and genuine ("ham") messages, thereby providing a robust barrier against unwanted communications. However, the success of machine learning models in detecting spam relies heavily on choosing and fine-tuning the features used for training [7], [8]. Challenges such as high dimensionality, feature redundancy, and the dynamic nature of spam content complicate feature selection [9]. Moreover, the interpretability of models is crucial for building trust and ensuring regulatory compliance, yet many advanced models operate as black boxes [10].

This research addresses these challenges by improving upon the work of SpotSpam [11], a recent approach that utilizes Support Vector Machines (SVM) combined with DistilBERT embeddings for SMS spam detection. While SVMs excel in high-dimensional spaces and work effectively with smaller, well-labeled datasets, their performance is highly contingent on meticulous hyperparameter tuning—a process that is both complex and experimental. To overcome this limitation, Optuna [12] is employed, an automatic hyperparameter optimization framework, to fine-tune the comparison model, achieving significant improvements in classification accuracy and overall model performance. Additionally, the impact of lemmatization during preprocessing is explored, with a comparison of models trained with and without this step. Subtle synonym data augmentation is applied to introduce

variability into the dataset, addressing the challenge of high dimensionality and feature redundancy.

To enhance model interpretability, LIME (Local Interpretable Model-Agnostic Explanations) [13] is employed. By providing transparent and interpretable explanations of the model's decisions, to ensure that the predictions are not only accurate but also explainable, contributing to greater transparency in the spam detection process.

The contributions of this research are as follows:

1) Hyperparameter Optimization: Applying Optuna [12] to fine-tune the hyperparameters of the comparison models.
2) Lemmatization Analysis: Analyzing the impact of lemmatization on model performance, comparing results with and without this preprocessing step to determine its effectiveness in reducing feature space complexity.
3) Model Explainability: This study employs LIME [13] to provide transparent and interpretable explanations of the model's decisions, contributing to both accuracy and explainability in spam detection.
4) The proposed model SVM+DistilBERT with Voting Classifier model achieves significant performance improvements over previous approaches like SpotSpam [11].

This research presents a comprehensive evaluation of machine learning models for SMS spam detection. Starting with foundational approaches such as SVM [14] [15], the study extends to more complex models, incorporating feature engineering techniques like TF-IDF vectorization [16] and ensemble classifiers such as XGBoost [17]. By integrating advanced embeddings like DistilBERT with SVM and optimizing hyperparameters using Optuna, the gap between traditional methods and modern advancements is bridged, enhancing precision and flexibility. This provides valuable insights for both academic researchers and industry practitioners seeking to develop effective and explainable spam detection systems.

The remainder of this paper is organized as follows: Section II presents the Literature Review. Section III provides a Detailed Description of the Methodology. Results and Analysis are presented in Section IV. Finally, Conclusions and Future Work are discussed in Section V.

II. Literature Review

Spam detection remains a critical task in the field of text classification, with numerous algorithms developed to address the challenge of accurately identifying unsolicited messages in datasets. Traditional machine learning methods, particularly Support Vector Machines (SVMs), have been extensively studied for this purpose.

Singh et al. [18] explored the use of SVM with TF-IDF and other feature extraction methods for SMS spam detection, demonstrating the effectiveness of SVM in such tasks. They noted the need for further comparative analyses of hybrid models and ensemble methods to enhance performance. Building on their findings, this study provides a detailed comparison of various SVM-based models, including combinations with advanced embeddings like DistilBERT [19], and evaluates the impact of preprocessing techniques such as lemmatization on model performance. By integrating LIME, this study also addresses the critical need for model explainability, enhancing transparency in the decision-making process. These contributions aim to provide a more nuanced understanding of SVM's potential in modern spam detection frameworks, aligning with the ongoing evolution of text classification techniques.

Almeida et al. [20] demonstrated the effectiveness of SVM classifiers in detecting spam within text messages by leveraging a comprehensive set of features extracted from the messages. Despite their success, they highlighted that the performance of SVMs is highly contingent on the meticulous selection and tuning of kernels and hyperparameters—a process that is both complex and experimental. Moreover, they pointed out potential scalability issues, as SVMs can become computationally intensive when applied to very large datasets.

In an effort to enhance SVM performance, Delany et al. [21] experimented with the integration of n-gram analysis. This hybrid approach improved spam detection rates by capturing contextual information within the text. However, the generation and processing of n-grams introduced additional computational overhead, leading to longer training times and increased memory usage. This trade-off suggests that while the method is robust, it may not be ideal for scenarios requiring immediate processing.

Text preprocessing and hyperparameter optimization are critical components in improving spam detection efficacy. Lemmatization, a preprocessing technique that reduces words to their base forms, can enhance traditional machine learning models like SVM by reducing feature space complexity and improving recall and precision. Akhmetov et al. [22] demonstrated the benefits of lemmatization across multiple languages, noting its importance in handling morphologically rich datasets.

The advent of transformer-based models has further revolutionized natural language processing tasks, including spam detection. Xiaoxu Liu et al. [23] demonstrated that models based on the vanilla transformer architecture perform well in SMS spam detection tasks. They suggested that utilizing more complex architectures like BERT could yield even better performance due to their ability to capture deeper contextual relationships with fewer features and ease of fine-tuning.

Despite these advancements, there is a notable gap in the literature concerning the integration of advanced embedding techniques with traditional machine learning models. Guzella et al. [24] reviewed the application of SVMs in spam filtering, highlighting their adaptability and accuracy. However, their work did not directly compare SVMs with emerging deep learning methods, which have shown potential for superior performance in text classification tasks.

To address this gap, the current research focuses on enhancing SVM models with modern embedding techniques such as DistilBERT. By combining the strengths of SVMs with the contextual understanding provided by advanced embeddings, the aim is to improve both the accuracy and adaptability of spam detection models. This research approach also involves evaluating various enhancements to traditional SVM models, including the use of term frequency-inverse document

frequency (TF-IDF), ensemble techniques, and preprocessing methods like lemmatization.

Considerations of model interpretability and complexity are crucial in the selection of appropriate spam detection methods. Metsis et al. [25] found that while SVMs generally outperform other machine learning algorithms in spam detection tasks, they suffer from a lack of interpretability compared to more transparent models like decision trees. Drucker et al. [26] attempted to enhance SVMs by incorporating boosting techniques, which improved accuracy but also introduced risks of overfitting and added complexity. These factors could hinder the deployment of such models in real-time prediction environments.

III. Methodology

The methodological approach focus is on the comparative analysis of various machine learning models for SMS Spam filtering purposes using the SVM based centric approach. The performance of these models will be evaluated based on their recall, accuracy, precision and the score of ROC AUC in correctly identifying spam messages. The flowchart of the process can be seen in Fig. 1.

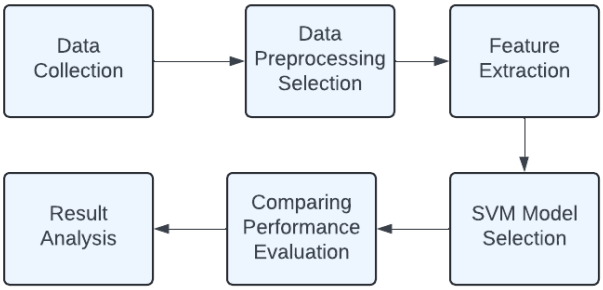

Fig. 1. Flowchart of the SVM comparison process.

A. Data Collection

The dataset utilized in this study is sourced from the UCI Machine Learning Repository [27], comprising 5,574 instances [28] with no missing values. Detailed information about the dataset composition and source distribution is presented in Table I.

TABLE I. Summary of SMS Message Sources for Spam Detection

Source	Description	Numb. of Messages
Grumbletext Website	UK forum reports of SMS spam.	425 spam
NUS SMS Corpus (NSC)	Legitimate messages from Singapore, mostly from students.	3,375 ham
Caroline Tag's PhD Thesis	Collection of SMS messages for research.	450 ham
SMS Spam Corpus v.0.1 Big	Combined collection of ham and spam messages for academic research.	1,002 ham, 322 spam

The bar chart in Fig. 2 revealed a significant imbalance in the dataset [28], with ham messages substantially outnumbering spam messages. To handle imbalance dataset, weight class balanced is employed on the SVM model. By completing these data preparation procedures, a well-organized and processed dataset is generated, which is now ready for the training and assessment of the classification models.

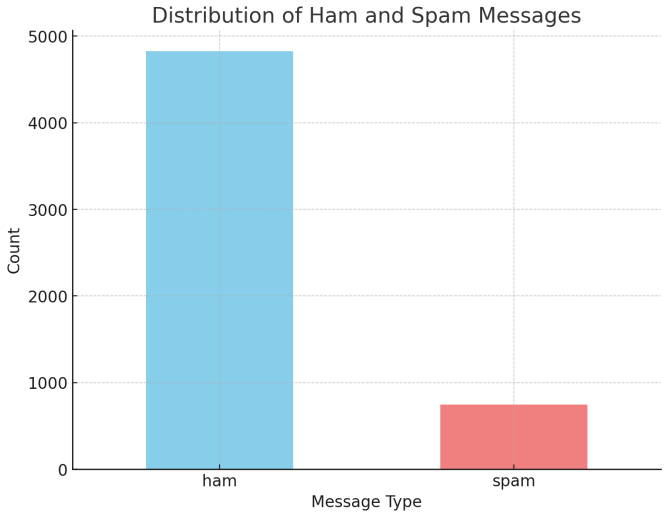

Fig. 2. Distribution of ham and spam messages in the dataset.

B. Data Preprocessing Selection

Before training the experimented models in this study, data cleaning is employed. For the SVM with DistilBERT embeddings, minimal cleaning is performed, involving converting the text to lowercase and removing extra white spaces. This approach retains most of the original raw data, as BERT embeddings are powerful enough to capture the semantic context.

To improve the generalization capability of the model, synonym data augmentation using WordNet was applied during the preprocessing stage for the proposed model. This technique introduces subtle variations in the text by replacing randomly selected words with their synonyms. The rationale behind this approach lies in the inherent diversity of language use in real-world communications, where different words can convey similar meanings. During synonym replacement, some words remain unchanged because WordNet may not have synonyms for them. The examples are shown in Table II.

While the application of synonym replacement in this study resulted in relatively minor changes to the text, it served two key purposes:

- Lexical Variety: The augmentation process exposed the model to variations in word usage that it might encounter in unseen data.
- Robustness to Minor Variations: Synonym replacement, despite introducing small changes, ensures that the model is less reliant on exact word matches.

In addition, this study investigated the influence of lemmatization on SMS spam detection models by preparing data in two different ways: with and without lemmatization. Lemmatization reduces words to their base forms (e.g. "congratulations" and "congrats" to "congratulate"), reducing the feature

space for model training. This preprocessing step enables for a comparative analysis of its impact on model performance.

TABLE II. COMPARISON OF ORIGINAL AND SYNONYM-REPLACED SMS MESSAGE

Original Text	**Text After Synonym Replacement**
Actually I decided I was too hungry so I haven't left yet : V	Actually I decided I was too hungry so I haven't **leave** yet : V
That's the thing with apes, u can fight to the death to keep something, but the minute they have it when u let go, that's it!	That's the thing with apes, u can fight to the death to keep something , but the **moment** they have it when u let go, that's it !
Glad to see your reply.	**glad** to see your reply.

C. Feature Extraction

1) TF-IDF: Term Frequency (TF) estimates the frequency of terms in a sentence by dividing the number of repetitions by the total number of words in the sentence. The IDF score determines the word's rarity within a corpus, suggesting that words that aren't used as frequently might hold more important information [29].

2) Bi-gram: The SVM TF-IDF will be enhanced with the bi-gram to capture more information context. A Bi-gram is two ceonsecutive elements which takes forms of words taken from the sequence of tokens. The bi-gram focuses on the word pair rather than capturing the meaning of the individual text itself.

For example, combining words like "customer service" is a bi-gram, which have more nuanced sentiment compared to an individual words such as "customer" or "service" [29].

3) DistilBERT: This study utilizes DistilBERT, a condensed version of the BERT (Bidirectional Encoder Representations from Transformers) model [30]. DistilBERT reduces the size of BERT by approximately 40%, making it more efficient in terms of memory and computational resources while retaining about 97% of BERT's performance on language understanding benchmarks.

DistilBERT achieves this efficiency through knowledge distillation [31], where a smaller student model learns to mimic the behavior of a larger teacher model. This process involves training the student model to reproduce the outputs of the teacher model, effectively capturing the essential knowledge in a more compact form without the need for extensive mathematical computations during inference.

The architecture of DistilBERT retains the Transformer-based design [32] but reduces the number of layers from 12 to 6. Despite this reduction, it maintains the ability to capture complex contextual relationships within the text through self-attention mechanisms. The self-attention mechanism allows the model to weigh the importance of different words in a sequence, enabling it to understand the context and nuances of language effectively. The architecture of DistilBERT can be seen in Fig. 3.

By integrating DistilBERT embeddings into the model, It leverages rich contextual representations of the input text, which enhances the performance of the spam detection task. This approach provides a balance between computational efficiency and model accuracy, making it suitable for applications requiring quick response times without significant loss in performance.

For detailed information on the mathematical formulations and training objectives of DistilBERT, readers are referred to Sanh et al. [19] and the foundational works on Transformers by Vaswani et al. [32].

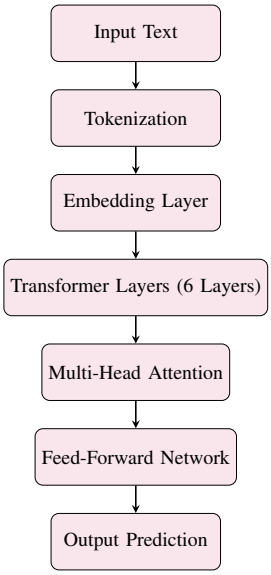

Fig. 3. Architecture of DistilBERT.

D. SVM Model Selection

1) Support Vector Machine: This study focuses on using Support Vector Machines (SVM) to filter spam in the SMS dataset. SVMs are supervised learning models used for regression analysis and classification. Gaye et al. (2021) [33] found that the SVM works by choosing the best hyperplane that separates the data into different classes. The main goal is to maximize the margin—the gap between the hyperplane and the nearest data points of each class—which can be either hard or soft. This separation challenge is transformed into a quadratic programming problem, allowing for the optimal hyperplane to be found efficiently.

This transformation is pivotal as it enables the SVM model to effectively handle linearly inseparable cases through the introduction of slack variables for soft margin optimization and the employment of kernel functions. Kernel functions, such as polynomial, radial basis function (RBF), and sigmoid, allow SVM to operate in high-dimensional spaces, facilitating the classification of complex datasets [34].

For the task of SMS spam filtering, the application of SVM is particularly promising due to its ability to discern between spam and legitimate messages with high precision. By constructing a feature vector from the SMS dataset and applying an appropriate kernel function, SVM can effectively classify

messages, leveraging the textual and contextual differences between spam and non-spam SMS. This capability is further evidenced by recent studies, which have demonstrated SVM's superior performance in text classification tasks compared to other machine learning algorithms [35] [36].

Furthermore, the adaptability of SVM in handling various types of data makes it an ideal choice for this study. By fine-tuning the SVM parameters, including the regularization parameter (C) and the kernel parameters, it can optimize the model to achieve maximum accuracy in spam detection. This optimization process is crucial for adapting the SVM model to the specific characteristics of the SMS dataset, thereby ensuring the effectiveness of the spam filtering solution.

2) AdaBoost: This study implements the ensemble classifier method AdaBoost for comparison analysis. AdaBoost is a powerful machine learning technique that combines multiple weak classifiers to create a robust, highly accurate classifier. It is simple to implement and relatively insensitive to noise in the data, but it can be affected by specific types of noise, such as class imbalance or outliers.

The following equation describes how each training instance's weight is updated by AdaBoost:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(X_i))}{Z_t} \tag{1}$$

In this equation, $D_t(i)$ represents the weight of the i-th training instance at iteration t, and α_t denotes the weight of the classifier. The exponential term adjusts the weight based on whether the prediction $h_t(X_i)$ matches the true label y_i. The normalization factor Z_t ensures that all weights sum to one, enabling the model to focus more on incorrectly classified instances [37].

In this study, AdaBoost with SVM is used as a comparison method to evaluate the performance of the proposed SVM-DistilBERT-based model and to assess the impact of lemmatization in classifying SMS spam messages.

3) eXtreme Gradient Boosting (XGBoost): XGBoost was employed in this study for comparative analysis alongside other models to evaluate the impact of lemmatization. XGBoost, or "Extreme Gradient Boosting", is a powerful and efficient machine learning algorithm built on the gradient boosting framework. Its high performance and speed make it popular for various supervised learning tasks. This algorithm is known for its scalability, sparsity awareness, and considerations for data compression, sharding, and cache-aware access [38].

These features enable XGBoost to efficiently manage large datasets with billions of entries, using fewer computational resources than many other systems [39]. In this study, it serves as a benchmark to understand how lemmatization affects model performance relative to other methods.

4) Voting Classifier: In this study, several machine learning models were combined with a voting classifier to create a reliable SMS spam detection system. The classifier combined Support Vector Machine (SVM), Random Forest, Gradient Boosting, Logistic Regression, and K-Nearest Neighbors into a single predictive model. The goal was to enhance the system's overall performance by using the advantages of each particular model.

The key to building an efficient SMS spam detection system is to balance the advantages of different machine learning models with each one's drawbacks. The ensemble approach seeks to use each model's robustness by combining these various models into a single voting classifier, producing a forecast that is more accurate and dependable. Every model makes a distinct contribution to the classifier; some are better at managing enormous datasets or offering interpretability, while others are better at handling high-dimensional data. Combining these models guarantees a more balanced approach to categorization that can handle the subtleties of SMS data with better accuracy while also improving the system's overall performance.

Support Vector Machine are very effective in a high dimensional data spaces and it is robust against outliers which is very good in handling noisy environments [40]. However, SVM also has its drawbacks that it faces scalability issues and is computationally intensive when handling a large datasets [41].

Random Forest(RF) can be integrated to help solve to address SVM drawbacks as they are very efficient in managing large datasets that leads to better generalization [41]. Random Forest is also very good in handling the missing data and conducting variable selection, however Random Forest may struggle on interpretability and imbalanced datasets [42] [43].

To add more strength to the ensemble, Logistic Regression were added to the Voting Classifier, that is known for it's simplicity and interpretability. It is very effective against binary classification problems and it is widely used for social sciences/medical field because of its interpretable and clear coefficients [44]. Logistic Regression can reduce the potential bias by handling the categorical predictor and continuous variables and also can effectively control confounding variables [45]. However, logistic regression has its drawbacks which is sensitive to outliers [46] and also has the assumption that there is a linear relationship between the predictors and the log-odds of the outcome that will effect on the limitation on its effectiveness for nonlinear boundaries [47].

On the other hand, Gradient Boosting model strength lies on it's high predictive accuracy and it's ability to adapt to the ensemble, especially when managing noisy data and multiple features [48] [49]. Gradient Boosting complements very good with the versatile capabilities of logistic regression and the robustness of Random Forest [50].

K-Nearest neighbor (KNN) contributes to the model because it's simple and effective to perform well in classification task, especially on how it groups the data based on the similarity. The flexibility of K-Neares neighbor (KNN) by leveraging different distance metrics improves its adaptability to different kinds of data [51]. When K-Nearest Neighbor (KNN) pairs with a structured models like Gradient Boosting and Random Forest, the combined model will perform better in handling complex data scenario [52].

The final output is generated by averaging the predictions made by each model in an ensemble technique. The voting classifier produces a more balanced and dependable detection

system, which integrates the outputs from SVM, Random Forest, Logistic Regression, Gradient Boosting, and K-nearest neighbors.

Using this method, the study shows that properly adjusted voting classifiers may greatly increase the precision and dependability of an SMS spam detection system.

E. Hyperparameter Optimization Using Optuna

In this study, Optuna is employed for hyperparameter optimization across various machine learning models for text classification, including XGBoost, AdaBoost, and Voting Classifiers with multiple base learners. To enhance readability and reduce complexity, standardized hyperparameter optimization across models were applied. For both XGBoost and AdaBoost, This study optimized the maximum number of features in the TF-IDF vectorizer, *max_features* $\in \{1000, 2000, 3000, 4000, 5000\}$, as well as the number of estimators, $n_{\text{estimators}} \in \{50, 100, 150, 200\}$, and the learning rate.

For XGBoost, additional parameters such as the maximum tree depth, *max_depth* $\in [3, 10]$, and the subsampling ratio, *subsample* $\in [0.5, 1.0]$, were optimized. Class imbalance was addressed using the *scale_pos_weight* parameter. In the Voting Classifier ensemble, shared hyperparameters across its constituent models were optimized to ensure consistency. The classifiers included SVC and Logistic Regression, where the regularization parameter C was optimized over a logarithmic scale. Additionally, Random Forest and Gradient Boosting classifiers were tuned for the number of estimators, with Random Forest also having an optimized maximum depth, and K-Nearest Neighbors varying in the number of neighbors.

When incorporating DistilBERT embeddings, the shared hyperparameters were optimized to maintain consistency across models. Stratified K-Fold cross-validation with $k = 5$ were employed, optimizing for metrics such as accuracy and ROC AUC. This systematic approach with Optuna enhanced model performance while reducing complexity and improving clarity in the hyperparameter optimization strategies.

F. Comparing Performance Evaluation

To evaluate the performance of the tested models, this study employ a technique called stratified k-fold cross-validation. Stratified K-fold cross-validation is an improved variant of k-fold cross-validation that is primarily used to ensure that the dataset's folds have similar class label distributions while maintaining the distribution of different classes [53]. With this method, over-fitting is reduced in datasets with imbalanced classes, unlike the regular k-fold cross validation, it can produce folds with skewd distribution of class labels in unbalanced datasets which may lead inaccurate performance measures.

After the integrated model is evaluated, the data is divided into five sections. Four of the segments are utilized to train the model, and the remaining one is used for testing. The aim of this assessment is to evaluate the model's differentiate capability across classes, with a specific focus on how well it can detect positive cases based on the ROC AUC score, Precision, Recall, F1-Score, and Accuracy.

G. Evaluation Measures

To provide different insights for the performance of classification model that were tested, this study includes different metrics such as Precision, Recall, F1-Score, ROC AUC Score and the Accuracy.

1) Precision: It is a statistical measures that evaluates how well a model predicts the favorable outcomes. It indicates the percentage of correctly predicted positive instances out of from all cases that were predicted positive. The notation for precision is denoted as in Eq. 2.

Mathematical Definition:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (2)$$

2) Recall: The proportion of accurately predicted positive observations to all of the observations made during the actual class is known as recall.

Mathematical Definition:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (3)$$

3) F1 score: The harmonic mean of Precision and Recall is the F1 score. It is helpful when attempting to achieve a balance between recall and precision.

Mathematical Definition:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

4) ROC AUC Score: A performance metric for classification issues at different threshold settings is the ROC (Receiver Operating Characteristic) AUC (Area Under the Curve) score. The degree or measure of separability is represented by AUC, and ROC is a probability curve. It indicates the degree to which the model can discriminate between classes.

Mathematical Definition: Plotting TPR (True Positive Rate, sometimes called Recall) against FPR (False Positive Rate) yields the ROC curve. The area under this curve, or AUC score, has the following mathematical definition:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \quad (5)$$

True Positive Rate (TPR) is defined as:

$$\text{TPR} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (6)$$

False Positive Rate (FPR) is defined as:

$$\text{FPR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \quad (7)$$

The AUC score falls between 0 and 1. An AUC of 1 indicates a perfect prediction model; an AUC of 0.5 indicates

a random prediction model. Better model performance is indicated by values nearer 1. More specifically, because it offers a thorough assessment of model performance across all classification thresholds, AUC is an important metric when assessing models on unbalanced datasets.

IV. Results and Analysis

The results in Table III and Table IV summarize various methods for SMS spam classification, including SVM-TF (Support Vector Machine with Term Frequency-Inverse Document Frequency), SVM-TF-Bi (SVM with TF-IDF and bi-grams), SVM-TF-Bi-Ada (SVM with TF-IDF Bi-gram and Adaboost), SVM-TF-Bi-XGB (SVM with TF-IDF Bi-gram and XGBoost), SVM-TF-Bi-Vote (SVM with TF-IDF Bi-gram and Voting Classifier), and SVM-DistilBERT-Vote (SVM with DistilBERT embeddings and Voting Classifier), where the "-Lem" suffix indicates the use of lemmatization, and performance is evaluated using the ROC AUC (Receiver Operating Characteristic Area Under Curve) metric.

TABLE III. Performance Comparison of Different Models: Models without Lemmatization

Model	ROC AUC	Precision	Recall	F1-Score	Accuracy
SVM-TF	0.9931	0.9878	0.8835	0.9327	0.9829
SVM-TF-Bi	0.9927	0.9719	0.8821	0.9248	0.9808
SVM-TF-Bi-Ada	0.9903	0.9984	0.8541	0.9206	0.9803
SVM-TF-Bi-XGB	0.9855	0.9387	0.8983	0.9179	0.9785
SVM-TF-Bi-Vote	0.9923	0.9984	0.8674	0.9283	0.9821
SVM-DistilBERT-Vote	0.9996	0.9973	0.9752	0.9861	0.9961

TABLE IV. Performance Comparison of Different Models: Models with Lemmatization

Model	ROC AUC	Precision	Recall	F1-Score	Accuracy
SVM-TF	0.9913	0.9855	0.9049	0.9433	0.9855
SVM-TF-Bi	0.9916	0.9856	0.9129	0.9477	0.9865
SVM-TF-Bi-Ada	0.9809	0.9803	0.7925	0.8764	0.9700
SVM-TF-Bi-XGB	0.9840	0.9512	0.8809	0.9145	0.9779
SVM-TF-Bi-Vote	0.9937	0.9925	0.8888	0.9376	0.9842
SVM-DistilBERT-Vote	0.9995	0.9951	0.9828	0.9878	0.9968

Table V compares the performance of this research with various previous studies on SMS spam classification, highlighting the differences in methods, accuracy, and datasets used. While traditional approaches such as TF-IDF with Random Forest [55] and XGBoost [59] reported accuracies of 97.50% and 97.64%, respectively, other methods like a hybrid system using K-Means SVM [56] and a Voting Classifier approach [57] achieved slightly higher accuracies, ranging from 98.8% to 98.93%. In contrast, this research, which employs an SVM DistilBERT model integrated with a Voting Classifier, achieves a superior accuracy of 99.6

The performance of various machine learning models on the SMS spam detection task was systematically evaluated. The results, summarized in Table III and Table IV, provide insights into how each model performed under two different preprocessing conditions: with and without the application of lemmatization. Lemmatization improves SMS spam identification by minimizing feature space redundancy, standardizing morphological variations (e.g. "running", "runs", and "ran" become "run"), and boosting generalization. For instance, a sample spam message, "Congrats! You've won a prize", is normalized to "congratulate! you win prize", allowing the algorithm to recognize spam-related phrases more accurately.

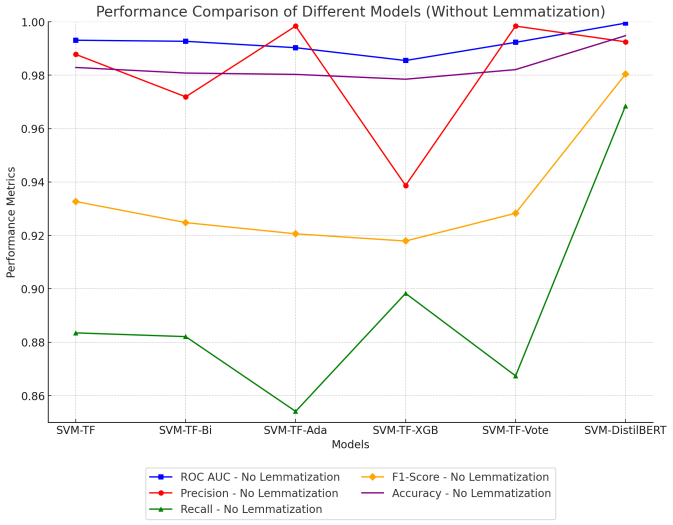

Fig. 4. Performance comparison of different models (without lemmatization).

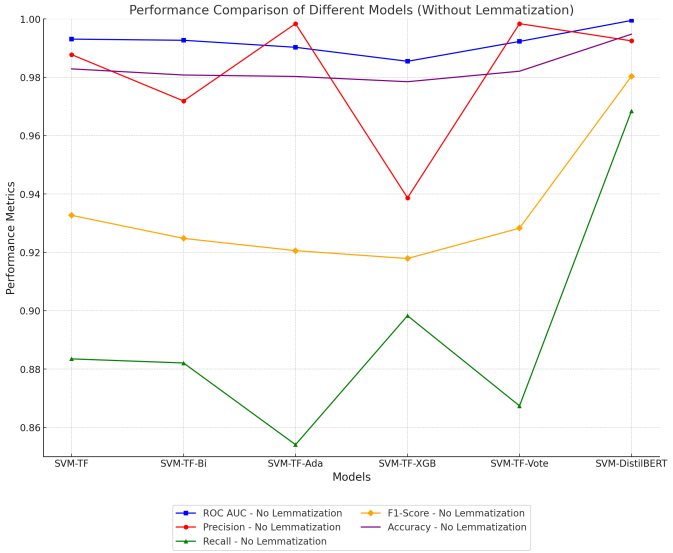

Fig. 5. Performance comparison of different models (with lemmatization).

This preprocessing step is especially beneficial for simpler models such as SVM-TF and SVM-TF-Bi, as evidenced by their higher recall and precision scores (Table IV). By emphasizing semantic meaning over lexical differences, lemmatization enhances detection accuracy and contributes to more effective spam filtering.

Fig. 5 and Fig. 4 showcase the performance metrics for six different models: SVM with Term Frequency (TF), SVM with TF and Bigrams (TF-Bi), SVM with TF and Bigram AdaBoost (TF-Bi-Ada), SVM with TF Bigram XGBoost (TF-Bi-XGB), SVM with TF Bigram and Voting Classifier (TF-Bi-Vote), and SVM with DistilBERT embeddings enhanced with Voting Classifier (SVM-DistilBERT-Vote). The key observation here is the effect of lemmatization on the models' performance.

Without lemmatization (Fig. 4), the ROC AUC remains consistently high across all models, with minor variations.

TABLE V. SUMMARY OF DIFFERENT METHODS IN CLASSIFYING SMS SPAM MESSAGES

Title and Reference	Dataset	Methods	Accuracy
SMS Spam Classification Using Machine Learning Techniques [54]	UCI Machine Learning Repository	SVM	98.797%
SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm [55]	UCI Machine Learning Repository	TF-IDF with Random Forest	97.50%
Hybrid SMS Spam Filtering System Using Machine Learning Techniques [56]	UCI Machine Learning Repository	K-Means SVM	98.8%
A Robust System For Message Filtering Using An Ensemble Machine Learning Supervised Approach [57]	6000 Messages Data, 1000 Messages are spam	Voting Classifier	98.93%
Semi-supervised novelty detection with one class SVM for SMS spam detection [58]	747 spam, 4827 non-spam messages	One class SVM	98%
Relevant SMS Spam Feature Selection Using Wrapper Approach and XGBoost Algorithm [59]	UCI Machine Learning Repository	XGBoost Classifier	97.64%
This Research	UCI Machine Learning Repository	SVM DistilBERT with Voting Classifier	99.6%

However, precision shows a significant drop for the SVM-TF-Ada model, indicating that this model might be struggling with false positives when lemmatization is not applied. Recall for the same model also dips notably, which could suggest that the model is less sensitive to actual spam messages without the normalization that lemmatization provides.

In contrast, with lemmatization applied (Fig. 5), the performance of the SVM-TF-Bi-Ada model sees a marked improvement in recall, indicating better detection of spam messages. The F1-Score, which balances precision and recall, reflects these changes, showing a more consistent performance across the models when lemmatization is used.

Interestingly, the SVM with DistilBERT embeddings integrated with Voting Classifier consistently shows high performance across all metrics, with and without lemmatization, suggesting that this model is robust to variations in text preprocessing. This robustness can likely be attributed to the sophisticated nature of the DistilBERT embeddings, which capture contextual information effectively even in raw, non-lemmatized text.

Overall, these results suggest that while lemmatization generally aids in improving recall and precision for certain models, particularly ensemble methods like AdaBoost and Voting Classifier, models leveraging advanced embeddings like DistilBERT are less dependent on such preprocessing steps. Therefore, the choice of preprocessing should be carefully considered depending on the model being used, with lemmatization being more crucial for traditional machine learning approaches.

1) Performance without Lemmatization: The SVM-DistilBERT-Vote model demonstrated superior performance across all evaluation metrics, positioning itself as the leading model in this task. Specifically, it achieved an ROC AUC of 0.9996, indicating near-perfect discrimination between spam and non-spam messages. The model also recorded a Precision of 0.9973, which reflects its high ability to correctly identify spam messages without including false positives. The Recall score of 0.9752 shows that the model effectively identified the majority of actual spam messages. This balance between high Precision and high Recall resulted in an F1-Score of 0.9861, underscoring the model's effectiveness in maintaining a low rate of both false positives and false negatives. The high Accuracy of 0.9961 further supports these findings, indicating that the model correctly classified a vast majority of messages.

Comparatively, other models, such as SVM-TF and various ensemble methods (SVM-TF-Bi-Vote, SVM-TF-Bi-XGB), also performed well but with some noticeable differences. For instance, the SVM-TF model achieved a Recall of 0.8835, lower than that of SVM-DistilBERT-Vote, suggesting that it missed more spam messages. Despite this, the SVM-TF model's Precision remained high at 0.9878, resulting in an F1-Score of 0.9327. While this score is robust, the model's lower Recall indicates a higher likelihood of misclassifying spam messages as non-spam, which could be critical in certain applications. The ensemble methods, while effective, exhibited similar trends, with generally strong Precision but lower Recall compared to SVM-DistilBERT-Vote, indicating a potential trade-off in these models between false positives and false negatives.

2) Performance with Lemmatization: The application of lemmatization yielded varying impacts across the different models, with the SVM-DistilBERT-Vote model once again demonstrating the highest performance. After lemmatization, the SVM-DistilBERT-Vote model maintained an ROC AUC of 0.9995, with only a slight reduction from the non-lemmatized version, which suggests that lemmatization had little effect on the model's ability to distinguish between classes. However, its Recall improved to 0.9828, leading to an F1-Score of 0.9878, slightly higher than without lemmatization. This improvement in Recall indicates that lemmatization helped the model better identify spam messages, making it even more reliable for practical use.

Other models, particularly those based on TF-IDF vectorization, showed more pronounced improvements with lemmatization. The SVM-TF-Bi model, for instance, experienced a notable increase in Recall from 0.8821 to 0.9129, which contributed to a rise in its F1-Score from 0.9248 to 0.9477. This suggests that lemmatization improved the model's ability to capture the semantic essence of the text, thereby reducing the number of missed spam messages. The enhancement in feature

representation due to lemmatization is likely responsible for this improvement.

However, not all models benefited from lemmatization. The SVM-TF-Bi-Ada model, in particular, saw a decline in performance with lemmatization, as evidenced by its decreased Recall (0.7925) and F1-Score (0.8764). This decline suggests that lemmatization may have disrupted the feature space that this ensemble method relies on, reducing its effectiveness in distinguishing between spam and non-spam messages. Such a decrease in performance highlights the importance of considering the model architecture when applying preprocessing techniques like lemmatization.

The variation in lemmatization's effect on models may arise from ensemble approaches such as AdaBoost, which depend on feature diversity that lemmatization might diminish, while simpler vectorization models gain from a less complex feature space.

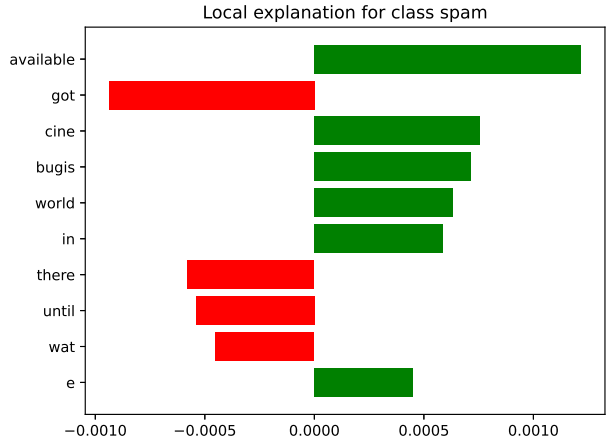

Fig. 6. Lime local variable explanation.

3) Explainable AI Using LIME: Fig. 6 presents a visual representation of explainable AI, generated using a tool called LIME (Local Interpretable Model-agnostic Explanations). This tool provides insights into how predictions are made by machine learning models through local explanations. LIME works by perturbing the input data—making slight alterations—and then assessing the impact of these modifications on the model's predictions.

In text classification tasks, such as the one illustrated in the figure, LIME identifies which words in a message have the most significant effect on the model's prediction. To enhance the interpretability of the model's decision-making process, it visualizes each word's contribution, highlighting those that increase or decrease the likelihood of the message being classified as spam.

For example, the word *"available"* has a contribution value of approximately **0.0010**, indicating a strong positive influence toward spam classification. On the other hand, *"got"* has a contribution of around **-0.0008**, meaning it significantly reduces the likelihood of the message being labeled as spam.

While certain words such as *"cine"* and *"bugis"* also contribute positively with values of **0.0006** and **0.0004** respectively, words like *"until"* and *"wat"* have smaller negative contributions, each near **-0.0004**. This numerical breakdown allows for a clearer understanding of which specific terms influence the spam classification and by how much, ensuring transparency in the model's decision-making process.

V. Conclusion

The study found that lemmatization often improves SMS spam detection performance, particularly in models that use TF-IDF vectorization. Lemmatization improves the model's ability to generalize and focus on message semantics by minimizing feature space redundancy and normalizing morphological variances. Models such as SVM-TF and SVM-TF-Bi showed considerable gains in recall and precision after lemmatization (Table IV), emphasizing its importance in increasing detection accuracy.

However, the impact of lemmatization is not uniform across all models. While models like SVM-TF-Bi showed enhanced performance with lemmatization, certain ensemble models, such as SVM-TF-Bi-Ada, experienced a decline, particularly in Recall and F1-Score. This suggests that the benefits of lemmatization are dependent on the specific model architecture and that its application should be considered carefully based on the characteristics of the model and the intended use case.

Overall, the findings underscore the importance of selecting the right preprocessing techniques in conjunction with the appropriate machine learning model to achieve optimal performance in text classification tasks. The variability in the effects of lemmatization across different models suggests that a one-size-fits-all approach may not be effective, and careful experimentation and analysis are necessary to determine the best preprocessing and modeling strategy for a given task.

Future works will be focusing on the different dataset other than the UCI Machine learning SMS Spam dataset that is more recent. Since this research is focusing on the SVM Centric to detect SMS Spam Detection, other methods that is more poweful while maintaing less computational cost still remains a challenge to address. Furthermore, synonym replacement in this study relied on WordNet without considering contextual similarity. Integrating models like BERT in the future could ensure replacements better align with sentence context, improving the quality of data augmentation.

Acknowledgment

The authors would like to express their sincere gratitude to King Abdulaziz University for the financial support provided through their scholarship program, which made this journal article possible. This support has been instrumental in enabling the research and completion of this work.

References

[1] Truecaller Insights. (2022) Truecaller insights 2022 us spam & scam report. Accessed: Insert Access Date. [Online]. Available: https://www.truecaller.com/blog/insights/truecaller-insights-2022-us-spam-scam-report

[2] S. Abdulhamid, M. S. A. Latiff, H. Chiroma, O. Osho, G. Abdul-Salaam, A. I. Abubakar, and T. Herawan, "A review on mobile sms spam filtering techniques," *IEEE Access*, vol. 5, pp. 15 650–15 666, 2017.

[3] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168 261–168 295, 2019.

[4] S. Rao, A. Verma, and T. Bhatia, "A review on social spam detection: Challenges, open issues, and future directions," *Expert Syst. Appl.*, vol. 186, p. 115742, 2021.

[5] T. Xia, "A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems," *IEEE Access*, vol. 8, pp. 82 653–82 661, 2020.

[6] A. P. Rodrigues, R. Fernandes, A. A, A. B, A. Shetty, A. K, K. Lakshmanna, and R. M. Shafi, "Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques," *Computational intelligence and neuroscience*, vol. 2022, p. 5211949, 2022, retraction published Comput Intell Neurosci. 2023 Oct 11;2023:9810910. [Online]. Available: https://doi.org/10.1155/2022/5211949

[7] H. Jain and R. K. Maurya, "A review of sms spam detection using features selection," *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, pp. 101–106, 2022.

[8] K. Zainal and M. Z. Jali, "A review of feature extraction optimization in sms spam messages classification," pp. 158–170, 2016.

[9] L. Zhang, "A new feature selection using dynamic interaction," *Pattern Analysis and Applications*, vol. 24, pp. 203–215, 2020.

[10] D. Bhusal, R. Shin, A. A. Shewale, M. K. M. Veerabhadran, M. Clifford, S. Rampazzi, and N. Rastogi, "Sok: Modeling explainability in security analytics for interpretability, trustworthiness, and usability," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ser. ARES '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3600160.3600193

[11] C. Oswald, S. E. Simon, and A. Bhattacharya, "Spotspam: Intention analysis–driven sms spam detection using bert embeddings," *ACM Trans. Web*, vol. 16, no. 3, Sep. 2022. [Online]. Available: https://doi.org/10.1145/3538491

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[14] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[15] D. A. Pisner and D. M. Schnyer, "Chapter 6 - support vector machine," in *Machine Learning*, A. Mechelli and S. Vieira, Eds. Academic Press, 2020, pp. 101–121. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000067

[16] S. D. Gupta, S. Saha, and S. K. Das, "Sms spam detection using machine learning," *Journal of Physics: Conference Series*, vol. 1797, no. 1, p. 012017, feb 2021. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1797/1/012017

[17] D. Jalal Mussa and N. G. M. Jameel, "Relevant sms spam feature selection using wrapper approach and xgboost algorithm," *KJAR*, vol. 4, no. 2, pp. 110–120, Nov. 2019.

[18] T. Singh, T. A. Kumar, and P. G. Shambharkar, "Enhancing spam detection on sms performance using several machine learning classification models," in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2022, pp. 1472–1478.

[19] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020. [Online]. Available: https://arxiv.org/abs/1910.01108

[20] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *ACM Symposium on Document Engineering*, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:13871930

[21] S. J. Delany, M. Buckley, and D. Greene, "Sms spam filtering: Methods and data," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9899–9908, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417412002977

[22] I. Akhmetov, A. Pak, I. Ualiyeva, and A. Gelbukh, "Highly language-independent word lemmatization using a machine-learning classifier," *Computación y Sistemas*, vol. 24, no. 3, pp. 1353–1364, 2020.

[23] S. Yerima and A. Bashar, "Semi-supervised novelty detection with one class svm for sms spam detection," *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, vol. CFP2255E-ART, pp. 1–4, 2022.

[24] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 206–10 222, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741740900181X

[25] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes - which naive bayes?" 01 2006.

[26] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[27] T. Almeida and J. Gómez Hidalgo, "SMS Spam Collection: A Public Set of SMS Labeled Messages," https://archive.ics.uci.edu/ml/datasets/sms+spam+collection, 2011, accessed: [25 November 2023].

[28] O. Abayomi-Alli, S. Misra, and A. Abayomi-Alli, "A deep learning method for automatic sms spam classification: Performance of learning algorithms on indigenous dataset," *Concurrency and Computation: Practice and Experience*, vol. 34, 2022.

[29] P. Joseph and S. Y. Yerima, "A comparative study of word embedding techniques for sms spam detection," in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022, pp. 149–155.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[33] B. Gaye, D. Zhang, and A. Wulamu, "Improvement of support vector machine algorithm in big data background," *Mathematical Problems in Engineering*, vol. 2021, p. 5594899, 2021. [Online]. Available: https://doi.org/10.1155/2021/5594899

[34] A. Tharwat, "Parameter investigation of support vector machine classifier with kernel functions," *Knowledge and Information Systems*, pp. 1–34, 2019.

[35] H. Lee and S. Kang, "Word embedding method of sms messages for spam message filtering," *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–4, 2019.

[36] C. Wang, Q. Li, T. Ren, X. Wang, and G. Guo, "High efficiency spam filtering: A manifold learning-based approach," *Mathematical Problems in Engineering*, 2021.

[37] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A comparative study of spam sms detection using machine learning classifiers," in *2018 eleventh international conference on contemporary computing (IC3)*. IEEE, 2018, pp. 1–7.

[38] N. Ghatasheh, I. Altaharwa, and K. Aldebei, "Modified genetic algorithm for feature selection and hyper parameter optimization: case of xgboost in spam prediction," *IEEE Access*, vol. 10, pp. 84 365–84 383, 2022.

[39] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[40] A. V. Messem, "Support vector machines: A robust prediction method with applications in bioinformatics," *Handbook of Statistics*, 2020.

[41] M. Singla and K. K. Shukla, "Robust statistics-based support vector machine and its variants: a survey," *Neural Computing and Applications*, vol. 32, pp. 11 173 – 11 194, 2019.

[42] T. Zhu, "Analysis on the applicability of the random forest," *Journal of Physics: Conference Series*, vol. 1607, 2020.

[43] M. N. Wright and I. König, "Splitting on categorical predictors in random forests," *PeerJ*, vol. 7, 2019.

[44] N. Gilbert, "Logistic regression," *Analyzing Tabular Data*, 2022.

[45] H. Nayebi, "Logistic regression analysis," *Advanced Statistics for Testing Assumed Casual Relationships*, 2020.

[46] I. A. I. Ahmed and W. Cheng, "The performance of robust methods in logistic regression model," *Open Journal of Statistics*, 2020.

[47] G. Zeng, "Logistic regression without intercept," *Asian Journal of Probability and Statistics*, 2022.

[48] C. Bentéjac, A. Csörgo, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2020. [Online]. Available: https://consensus.app/papers/analysis-gradient-boosting-algorithms-bentjac/c3be7a8bb964590488d56f7b829a39ab/?utm_source=chatgpt

[49] A. Ustimenko, L. Prokhorenkova, and A. Malinin, "Uncertainty in gradient boosting via ensembles," *ArXiv*, vol. abs/2006.10562, 2020. [Online]. Available: https://consensus.app/papers/uncertainty-gradient-boosting-ensembles-ustimenko/dbda713e90385394afce8d98dc7ac077/?utm_source=chatgpt

[50] D. Schalk, B. Bischl, and D. Rugamer, "Accelerated componentwise gradient boosting using efficient data representation and momentum-based optimization," *ArXiv*, vol. abs/2110.03513, 2021. [Online]. Available: https://consensus.app/papers/accelerated-componentwise-gradient-boosting-using-schalk/03898648792956a8811614558f4b9abe/?utm_source=chatgpt

[51] S. Zhang, "Challenges in knn classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 4663–4675, 2021.

[52] A. Shokrzade, M. Ramezani, F. Tab, and M. A. Mohammad, "A novel extreme learning machine based knn classification method for dealing with big data," *Expert Syst. Appl.*, vol. 183, p. 115293, 2021.

[53] J. A. Sáez and J. L. Romero-Béjar, "Impact of regressand stratification in dataset shift caused by cross-validation," *Mathematics*, 2022. [Online]. Available: https://consensus.app/papers/impact-regressand-stratification-dataset-shift-caused-sez/b32921947c5753ed9fb8bd1090b79fc4/?utm_source=chatgpt

[54] T. Jain, P. Garg, N. Chalil, A. Sinha, V. K. Verma, and R. Gupta, "Sms spam classification using machine learning techniques," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022, pp. 273–279.

[55] N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "Sms spam message detection using term frequency-inverse document frequency and random forest algorithm," *Procedia Computer Science*, vol. 161, pp. 509–515, 2019, the Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919318617

[56] H. Baaqeel and R. Zagrouba, "Hybrid sms spam filtering system using machine learning techniques," in *2020 21st International Arab Conference on Information Technology (ACIT)*, 2020, pp. 1–8.

[57] A. Mahabub, M. Innat, and M. Faruque, "A robust system for message filtering using an ensemble machine learning supervised approach," *ICIC Express Letters*, vol. 10, pp. 805–811, 07 2019.

[58] S. Yerima and A. Bashar, "Semi-supervised novelty detection with one class svm for sms spam detection," *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, vol. CFP2255E-ART, pp. 1–4, 2022.

[59] D. Mussa and N. M. Jameel, "Relevant sms spam feature selection using wrapper approach and xgboost algorithm," *Kurdistan Journal of Applied Research*, vol. 4, pp. 110–120, 11 2019.