Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

Optimizing SMS Spam Detection Using Machine Learning: A Comparative Analysis of Ensemble and Traditional Classifiers

Gregorius Airlangga[1)*]
[1)]Atma Jaya Catholic University of Indonesia
[1)]gregorius.airlangga@atmajaya.ac.id

ABSTRACT

With the rapid rise of mobile communication, Short Message Service (SMS) has become an essential platform for transmitting information. However, the growing volume of unsolicited and harmful spam messages presents significant challenges for both users and mobile network operators. This study explores the effectiveness of various machine learning models, including Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, and an Ensemble Voting Classifier, in detecting SMS spam. A dataset containing 5,572 SMS messages, labeled as either spam or ham (legitimate), was used to evaluate these models. Hyperparameter tuning was performed on each model to optimize accuracy, and the models were assessed using metrics such as precision, recall, F1-score, and accuracy. The results indicated that the SVM and Ensemble Voting Classifier achieved the highest performance, with accuracies of 0.9857 and 0.9848, respectively. Both models demonstrated superior recall for spam messages, making them highly effective for real-world spam detection systems. While Random Forest, Gradient Boosting, and AdaBoost also performed well, their slightly lower recall for spam suggests that they may misclassify some spam as legitimate messages. The study highlights the effectiveness of machine learning models in addressing the SMS spam problem, particularly when using ensemble methods. Future research should focus on addressing class imbalance and exploring deep learning approaches to further enhance model performance. These findings offer valuable insights for developing more accurate and scalable SMS spam detection systems.

Keywords: SMS spam detection; Machine learning; Ensemble classifiers; Support Vector Machine; Spam classification

1. INTRODUCTION

The rise of mobile communication technologies has brought about significant advancements in how people connect, and Short Message Service (SMS) remains one of the primary methods of direct communication, especially in regions with limited internet access (Božanić & Sinha, 2021; Ling et al., 2020; Rida, 2021). However, with the widespread use of SMS, there has been a surge in unsolicited and harmful messages commonly referred to as spam (Maqsood et al., 2023; Patil et al., 2022; Sharaff et al., 2021). These spam messages range from unwanted advertisements to more malicious threats, such as phishing scams and malware distribution, posing risks not only to users but also to the security of mobile networks (Weichbroth Paweł and Łysik, 2020). Detecting and mitigating SMS spam is increasingly crucial, especially as traditional detection methods fail to keep pace with the sophistication of modern spam. As a result, machine learning has emerged as a promising solution for building more effective and adaptable spam detection systems (Jáñez-Martino, Alaiz-Rodr'iguez, González-Castro, Fidalgo, & Alegre, 2023). In the early stages of SMS spam detection research, rule-based systems were the dominant approach (Saidani et al., 2020). These systems relied on predefined rules such as the presence of specific keywords or patterns in the message content to classify spam (Jáñez-Martino et al., 2023). While rule-based approaches provided initial success, they quickly became outdated as spammers devised new ways to avoid detection (Rao et al., 2021). This led to a shift towards machine learning-based methods, which are capable of learning from data and automatically detecting spam without the need for manually defined rules.

Naive Bayes classifiers were among the first machine learning techniques applied to spam detection. Known for their simplicity and effectiveness in text classification, Naive Bayes models compute the probability of a message being spam based on the occurrence of specific words (Daisy & Begum, 2021). However, despite their initial success, Naive Bayes models struggle with imbalanced datasets, which are common in spam detection tasks, where legitimate messages vastly outnumber spam (Bose, 2023). Support Vector Machines (SVM) also gained popularity in SMS spam detection due to their ability to handle high-dimensional data. SVMs aim to find an optimal boundary that separates

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

spam from non-spam messages (Choi et al., 2024). Although SVMs can achieve good accuracy, their computational cost, especially when working with large datasets, makes them less practical for real-time applications (Afifi et al., 2020).

More recent research has focused on ensemble learning techniques, which combine multiple classifiers to improve performance (Ganaie et al., 2022). Random Forest, Gradient Boosting, and AdaBoost are popular ensemble methods used for spam detection (Fayaz et al., 2020). These models operate by aggregating the predictions of several individual models, reducing overfitting and increasing robustness. For example, Random Forest builds an ensemble of decision trees and averages their predictions to improve accuracy (Zhou et al., 2020). Gradient Boosting constructs models sequentially, where each new model corrects the errors of its predecessor, resulting in a strong predictive model. The rapid increase in the volume and complexity of spam messages poses serious risks to both individuals and organizations (Zhou et al., 2020). Spam messages disrupt communication, degrade the user experience, and in more severe cases, lead to financial loss or data breaches. Particularly in developing regions where SMS is a primary means of communication, users are at greater risk of being deceived by phishing schemes, fraudulent requests, or malware distributed through spam messages (Jáñez-Martino et al., 2023). Furthermore, mobile network operators bear the financial burden of delivering spam messages, which adds to the urgency of developing robust spam detection systems.

The growing sophistication of spammers, who now use techniques such as message obfuscation, URL shortening, and content masking, exacerbates the problem. Traditional spam detection systems, including keyword-based filters and manual reporting, are increasingly ineffective against these tactics (Alkhalil et al., 2021). Therefore, there is an urgent need for a scalable, accurate, and adaptive system that can effectively detect and prevent spam in real-time. Modern SMS spam detection techniques are heavily reliant on machine learning models that can process and classify textual data (Tusher et al., 2024). The most widely used method for transforming SMS messages into numerical data for machine learning models is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF helps identify the relative importance of words within a message, making it easier for models to differentiate between spam and legitimate messages (Islam et al., 2021).

In terms of classifiers, ensemble models such as Random Forest and Gradient Boosting represent the state of the art. Random Forest, which builds an ensemble of decision trees and averages their predictions, is highly effective in handling noisy data and mitigating overfitting (Zhang et al., 2023). Gradient Boosting, which builds models in a sequential manner to correct errors made by previous models, is another leading method due to its ability to refine its predictions over time. Support Vector Machines (SVM) remain a strong candidate for SMS spam detection due to their effectiveness in high-dimensional data classification (Saidani et al., 2020). However, their computational complexity, particularly with large datasets, limits their applicability in real-time systems. Ensemble techniques, such as Voting Classifiers, which combine the outputs of multiple base classifiers, have gained significant attention. By taking the majority vote or averaging the probabilities predicted by individual models, Voting Classifiers provide more robust predictions and improve accuracy, especially in cases where no single model consistently outperforms others (Mushtaq et al., 2022).

Despite advancements in machine learning techniques for SMS spam detection, there are still challenges that have yet to be fully addressed. One major challenge is the imbalance between spam and legitimate messages in most datasets. This imbalance can cause machine learning models to favor the majority class, leading to poor performance in detecting spam messages (Le Jeune et al., 2021). Various methods, such as the Synthetic Minority Over-sampling Technique (SMOTE), have been proposed to address this issue, but further research is needed to find the most effective ways of improving performance on imbalanced datasets (Alam et al., 2022). Another limitation of current spam detection systems is the lack of interpretability (Abu-Salih et al., 2022). Many machine learning models, such as Random Forest and Gradient Boosting, are often treated as black boxes, making it difficult to understand how they arrive at their predictions. In practical applications, especially in security-sensitive domains, the ability to explain a model's decisions is crucial for building trust. While there are some techniques for feature importance analysis, these are not widely used in SMS spam detection and require more attention. Lastly, the dynamic nature of spam poses a continuous challenge. Spammers constantly adapt their tactics to bypass detection systems, using techniques like URL obfuscation and content manipulation (Kulkarni et al., 2024). Although machine learning models can adapt to new data, more research is needed to develop models that are both adaptable and scalable enough to handle real-time spam detection in evolving environments.

The goal of this research is to address these gaps by developing a more advanced system for SMS spam detection that leverages multiple machine learning models and ensemble methods. Specifically, this study will compare the performance of five classifiers: Random Forest, Gradient Boosting, AdaBoost, SVM, and Logistic Regression and explore the benefits of using an Ensemble Voting Classifier. The aim is to assess which model or combination of

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

models can provide the best trade-off between accuracy, computational efficiency, and real-world applicability for spam detection. This research will also investigate the use of advanced resampling techniques to handle imbalanced datasets and improve the interpretability of the models through feature importance analysis. The focus will be on building a system that is not only highly accurate but also interpretable and adaptable to the evolving nature of spam.

The rest of this article is organized as follows: The next section describes the data set used in the study and details the preprocessing techniques applied to prepare the data for model training. The following section outlines the machine learning models used, including the hyperparameter tuning process and cross-validation setup. Afterward, the results section presents the performance metrics of the models, including accuracy, precision, recall, and AUC-ROC curves. The discussion section interprets these findings in the context of existing research, with a focus on model interpretability, handling data imbalance, and adaptability to new spam patterns. The conclusion summarizes the key contributions of the study and suggests directions for future research in SMS spam detection.

LITERATURE REVIEW

The task of detecting SMS spam has undergone significant evolution over the years, progressing from simple rule-based systems to sophisticated machine learning approaches (Kulkarni et al., 2024). The early stages of SMS spam detection focused on manually crafted rules, relying on keyword detection and pattern matching. These rule-based systems, though effective in initial spam detection tasks, struggled to adapt to the increasing complexity and diversity of spam messages as spammers developed new tactics (Tusher et al., 2024). As spam techniques evolved, rule-based systems required constant updates, and the rigid nature of these systems hindered their scalability. The limitations of rule-based systems led to a shift towards statistical methods, with Naive Bayes being one of the earliest machine learning techniques applied to SMS spam detection. Naive Bayes classifiers, as explored in studies by (Daisy & Begum, 2021), assume that features (in this case, words or tokens in a message) are independent of each other. Although Naive Bayes is computationally efficient and performs well in text classification tasks, it often struggles with imbalanced datasets, where legitimate messages significantly outnumber spam messages (de Zarzà et al., 2023). This imbalance can cause Naive Bayes models to underperform, particularly when dealing with rare but important spam patterns. In addition, Naive Bayes models assume that the presence of one feature is unrelated to the presence of another, which is not always the case in real-world data, limiting their effectiveness in more complex spam detection tasks (Noekhah et al., 2020).

Support Vector Machines (SVM) have also been widely used in SMS spam detection due to their ability to handle high-dimensional data. SVM models, as noted by (Gaye et al., 2021), are particularly effective at creating a clear separation between classes by identifying an optimal hyperplane in feature space. The strength of SVM lies in its ability to handle non-linear data when combined with kernel functions, making it a robust option for spam classification. However, SVMs are computationally expensive, especially when working with large datasets, and require careful tuning of parameters such as the regularization term and kernel type. This complexity makes SVM less practical for real-time spam detection in large-scale environments. With advancements in ensemble learning, techniques such as Random Forest, Gradient Boosting, and AdaBoost have become state-of-the-art methods for SMS spam detection. Random Forest, introduced by (Genuer et al., 2020), builds an ensemble of decision trees by randomly selecting subsets of features and samples to train each tree. This approach mitigates the problem of overfitting, which is common in single decision trees, and results in more robust performance across diverse datasets. In spam detection, Random Forest has demonstrated superior performance due to its ability to handle large feature spaces and noisy data. Studies such as those by (Shaaban et al., 2022) highlight Random Forest's effectiveness in text classification tasks, including SMS spam detection.

Gradient Boosting, another powerful ensemble method, builds models sequentially, with each new model correcting the errors of the previous ones. Introduced by (Prosise, 2022), Gradient Boosting has shown exceptional performance in a variety of classification tasks. Its ability to improve iteratively makes it a strong candidate for handling complex datasets, including those with subtle spam patterns that are harder to detect. While Gradient Boosting is effective, it can be prone to overfitting if not carefully regularized, and its sequential nature can result in slower training times compared to other models like Random Forest. AdaBoost, developed by (Mienye & Sun, 2022), is another ensemble technique that focuses on combining weak learners (often decision trees with a single split, known as stumps) to create a strong classifier. AdaBoost assigns higher weights to misclassified instances during each iteration, allowing it to focus on harder-to-classify messages. Although AdaBoost has been effective in many spam detection tasks, it is sensitive to noisy data and outliers, which can negatively impact its performance (Maurya et al., 2023).

The rise of ensemble techniques such as Voting Classifiers has marked a shift towards combining the strengths of multiple models to improve classification performance. Voting Classifiers aggregate the predictions of different base

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

models (e.g., Random Forest, Gradient Boosting, SVM) and either take a majority vote (hard voting) or average the predicted probabilities (soft voting) to make a final prediction (Awe et al., 2024). This approach, as discussed by (Roy et al., 2020), leverages the strengths of individual classifiers while compensating for their weaknesses, resulting in more robust and accurate spam detection systems. While machine learning models have demonstrated their effectiveness in detecting SMS spam, they are not without limitations. One of the most significant challenges in spam detection is dealing with imbalanced datasets, where legitimate messages far outnumber spam. This imbalance can lead to biased models that are more likely to misclassify spam messages as legitimate. Several techniques have been proposed to address this issue, such as Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples for the minority class to balance the dataset. Studies by (Abid et al., 2022) have shown that SMOTE can significantly improve the performance of classifiers in imbalanced datasets, but its effectiveness in SMS spam detection still requires further exploration.

Additionally, the interpretability of machine learning models is a growing concern in SMS spam detection. Models like Random Forest and Gradient Boosting, while achieving high accuracy, are often treated as black boxes, making it difficult to understand how they arrive at their predictions (Carmona et al., 2022) In practical applications, especially in areas such as mobile security and fraud detection, being able to explain a model's decision is crucial for building trust with users and stakeholders (Dhieb et al., 2020). Techniques such as feature importance analysis have been introduced to address this challenge, providing insights into which features (e.g., specific words or phrases) are most relevant for identifying spam. However, interpretability remains an underexplored area in the context of SMS spam detection, with many models still lacking transparency. Moreover, spam messages are constantly evolving. Spammers use sophisticated techniques such as message obfuscation, URL shortening, and content masking to bypass detection systems (Swarnkar et al., 2022). Machine learning models, while effective at detecting known spam patterns, often struggle to keep pace with these evolving threats. Continuous learning and adaptability are essential for the development of future spam detection systems, but there is still a gap in research exploring how models can be made more adaptive to new forms of spam.

Based on these literature reviews, from early rule-based systems to the more advanced machine learning and ensemble methods of today, SMS spam detection has made significant strides. However, as spammers continue to develop more sophisticated techniques, challenges such as data imbalance, model interpretability, and adaptability remain. While ensemble models such as Random Forest, Gradient Boosting, and Voting Classifiers have shown promise in addressing these issues, more research is needed to optimize these models for real-time, scalable, and interpretable spam detection. This study builds upon the existing literature by comparing multiple machines learning models, including ensemble techniques, and addressing key challenges such as data imbalance and interpretability

METHOD

The research methodology employed in this study involves multiple phases, including dataset preparation, feature extraction, model development, hyperparameter optimization, and performance evaluation. The following sections detail the processes as presented in figure 1. This approach allows for a cohesive understanding of the SMS spam detection system's construction and evaluation. The dataset for this research consists of SMS messages, where each message is labeled as either "spam" or "ham" (legitimate). Let $(D = \{(x_i, y_i)\}_{i=1}^{n})$ represent the dataset, where $(x_i \in R^d)$ denotes the feature vector corresponding to the (i)-th SMS message, and $(y_i \in \{0,1\})$ is the binary label, with 0 representing ham and 1 representing spam. The goal is to develop models capable of predicting the label (y_i) based on the content of the message (x_i).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

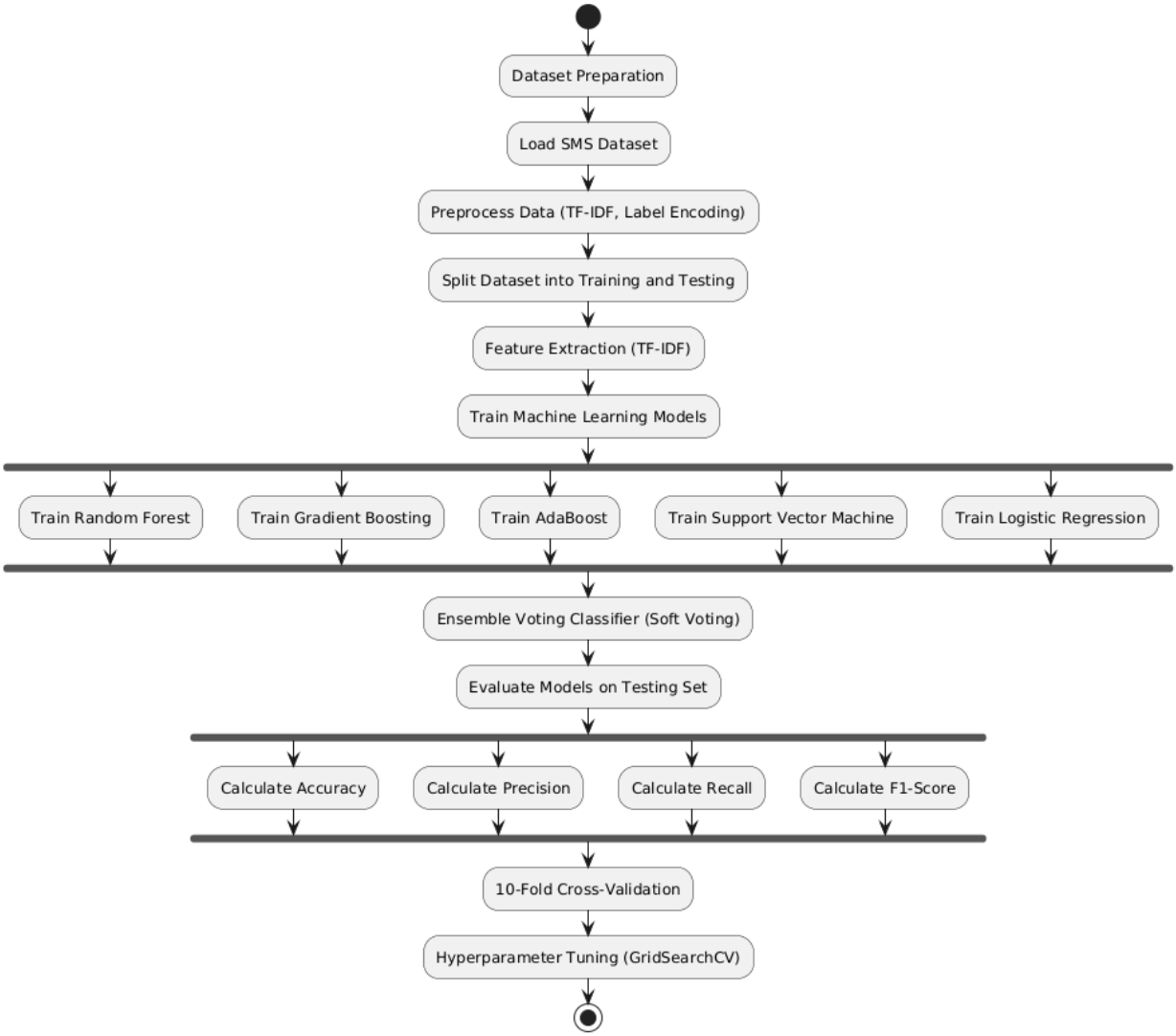

Figure 1. Data Science Pipeline

The text data requires preprocessing to convert the messages into a format suitable for machine learning models. The first step is transforming the raw text data into numerical vectors. This transformation is performed using the Term Frequency-Inverse Document Frequency (TF-IDF) method, which captures the importance of words within a message relative to their occurrence across all messages. Mathematically, the TF-IDF score for a term (t) in a message (x_i) is calculated as the product of the term frequency (TF) and the inverse document frequency (IDF). The TF component is the count of term (t) in message (x_i), normalized by the total number of terms in (x_i). The IDF component is calculated as $\text{IDF}(t) = \log\left(\frac{n}{1+|\{x_i|t \in x_i\}|}\right)$, where (n) represents the total number of messages, and ($|\{x_i \mid t \in x_i\}|$) denotes the number of messages containing term (t). The TF-IDF transformation results in a matrix ($V \in R^{n \times d}$), where each row corresponds to a message and each column represents the TF-IDF value of a term.

Label encoding is applied to the target variable (y_i), converting the labels into a numerical format, with 0 representing ham and 1 representing spam. After preprocessing, the dataset is split into training and testing subsets, maintaining the original class distribution. Denote the training set as ($D\_\{Ir\{train\}\} \subset D$) and the testing set as ($D_{\text{test}} \subset D$), such that ($D_{\text{train}} \cup D_{\text{test}} = D$) and ($D_{\text{train}} \cap D_{\text{test}} = \emptyset$).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

This study evaluates five machine learning classifiers: Random Forest (RF), Gradient Boosting (GB), AdaBoost (AB), Support Vector Machine (SVM), and Logistic Regression (LR). Additionally, an Ensemble Voting Classifier (VC) is constructed by combining the predictions of these base models. Each model is trained on the training set (D_{train}) and evaluated on the testing set (D_{test}). The Random Forest classifier is an ensemble of decision trees. Each tree ($h_j(x)$) is trained on a randomly sampled subset of the features and samples from (D_{train}). The final prediction ($\hat{y}$) for an input (x) is obtained by averaging the predictions of all trees $\hat{y} = \frac{1}{m}\sum_{j=1}^{m} h_j(x)$, where (m) is the number of trees in the forest. The model's hyperparameters include the number of trees (m), the maximum depth of each tree, and the minimum number of samples required to split a node. These parameters are tuned to optimize the model's performance.

Gradient Boosting builds sequential models, where each new model corrects the errors made by the previous ones. The model updates are guided by minimizing a loss function ($\mathcal{L}(y, \hat{y})$), where (y) is the true label, and ($\hat{y}$) is the predicted label. At each iteration, the model adjusts its prediction as follows $h_t(x) = h_{t-1}(x) + \eta \cdot g_t(x)$ where (η) is the learning rate and ($g_t(x) = -\nabla_{\hat{y}}\mathcal{L}(y, \hat{y})$) represents the gradient of the loss function with respect to the predictions. The iterative process allows Gradient Boosting to focus on the hardest-to-classified examples. Furthermore, adaBoost constructs a strong classifier by iteratively combining weak learners. At each iteration (t), the model assigns higher weights to misclassified examples, making them more important in subsequent iterations. The weight for the (i)-th sample at iteration ($t+1$) is updated according to the following rule $w_i^{(t+1)} = w_i^{(t)} \cdot \exp\left(\alpha_t \cdot I\left(y_i \neq \widehat{y_i^{(t)}}\right)\right)$ where (α_t) is the weight assigned to the weak learner at iteration (t), computed as $\alpha_t = \frac{1}{2}\log\left(\frac{1-e_t}{e_t}\right)$, and ($e_t$) is the error rate at iteration (t). The final classifier is a weighted sum of all the weak learners. Besides, Support Vector Machines (SVM) aim to find a hyperplane that maximally separates spam and ham messages in the feature space. Given a set of training examples ($\{(x_i, y_i)\}$), the SVM solves the following optimization problem $\min_{w,b} \frac{1}{2}|w|^2 \quad \text{subject to} \quad y_i(w^{\top}v_i + b) \geq 1, \forall I$ where (w) is the weight vector, and (b) is the bias term. The SVM classifier uses a kernel function to map the input data into a higher-dimensional space, allowing it to learn non-linear decision boundaries.

Logistic Regression models the probability of a message being spam using a logistic function. The probability ($p(y = 1 \mid v_i)$) is given by $p(y = 1 \mid v_i) = \frac{1}{1+\exp(-w^{\top}v_i)}$ where (w) is the vector of model coefficients, learned by maximizing the log-likelihood of the training data. The Ensemble Voting Classifier combines the predictions of the five base models such as Random Forest, Gradient Boosting, AdaBoost, SVM, and Logistic Regression using soft voting. In soft voting, the final prediction ($\hat{y}$) is the class with the highest average predicted probability $\hat{y} = \arg\max_k \frac{1}{T}\sum_{t=1}^{T} P_t(y = k \mid x)$ where (T) is the number of base classifiers, and ($P_t(y = k \mid x)$) is the predicted probability of class (k) by classifier (t).

Model performance is evaluated using several metrics, including accuracy, precision, recall, and F1-score. Accuracy is calculated as $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$, where (TP), (TN), (FP), and (FN) represent the number of true positives, true negatives, false positives, and false negatives, respectively. Precision is given by $\text{Precision} = \frac{TP}{TP+FP}$, and recall is calculated as $\text{Recall} = \frac{TP}{TP+FN}$. The F1-score is the harmonic mean of precision and recall, defined as $\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$. To ensure generalizability, 10-fold cross-validation is applied. The dataset is randomly divided into 10 equal subsets, and each model is trained on 9 subsets while being tested on the remaining subset. The process is repeated 10 times, and the average performance across all folds is reported. Hyperparameter tuning is conducted using GridSearchCV, where a grid of predefined hyperparameters is tested to find the optimal configuration that maximizes model performance. This methodology is designed to ensure rigor and robustness in the evaluation of the models, with a focus on both predictive accuracy and generalizability.

RESULT & DISCUSSIONS

This section provides a comprehensive analysis of the results obtained from the various machine learning models used in this study for SMS spam detection. The models evaluated include Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, and an Ensemble Voting Classifier as presented in table 1. Each model was tuned using hyperparameter optimization, and its performance was assessed based on metrics

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

such as accuracy, precision, recall, and F1-score. These metrics provide a clear understanding of how well each model performs in distinguishing between ham (legitimate) and spam messages. The dataset consists of 5,572 SMS messages, of which 4,827 are labeled as ham and 745 as spam and can be download from (Dapat, 2024).

This class imbalance, where legitimate messages significantly outnumber spam, presents a challenge for accurate spam classification. The models were evaluated using a stratified train-test split, ensuring that both the training and testing sets maintain the original class distribution. The process of hyperparameter tuning was performed using GridSearchCV to optimize each model for maximum accuracy. This allowed fine-tuning key parameters, resulting in the following optimized configurations for each model. The Random Forest classifier achieved its best performance with 100 trees, no restriction on maximum depth, and a minimum of five samples required to split a node.

Results

Gradient Boosting reached optimal results with a learning rate of 0.1, a maximum depth of 5, and 200 boosting stages. AdaBoost used 100 estimators and a learning rate of 1.0, while Support Vector Machine performed best with a linear kernel, regularization parameter ($C = 10$), and the 'scale' gamma setting. Logistic Regression was tuned with ($C = 10$), the L2 penalty, and the 'lbfgs' solver. Upon evaluating these models on the test set, which contains 1,115 messages, the following results were observed. The Random Forest classifier achieved a test accuracy of 0.9740. The precision, recall, and F1-score for ham messages were 0.97, 1.00, and 0.99, respectively, indicating the model's strong ability to correctly identify ham messages. However, for spam, while the precision was perfect at 1.00, the recall dropped to 0.81, leading to an F1-score of 0.89. This indicates that the model occasionally misclassified spam as ham, reducing the effectiveness of its recall for spam detection.

Gradient Boosting achieved a similar accuracy of 0.9713. For ham messages, the precision, recall, and F1-score were 0.97, 0.99, and 0.98, respectively. The performance on spam messages was slightly lower, with precision at 0.96 and recall at 0.82, resulting in an F1-score of 0.88. Like Random Forest, Gradient Boosting struggled with recall for spam, meaning that some spam messages were incorrectly classified as legitimate. The AdaBoost classifier also yielded an accuracy of 0.9713. Its performance on ham messages remained strong, with precision and recall both near 0.97 and 0.99. However, the spam detection precision was 0.95, and the recall was 0.83, leading to an F1-score of 0.88 for spam. Although the precision was slightly lower than the other models, it still exhibited similar trends with a decline in recall for spam.

The Support Vector Machine classifier outperformed the other individual models, with a test accuracy of 0.9857. The precision, recall, and F1-score for ham were 0.98, 1.00, and 0.99, respectively, showing nearly perfect classification of legitimate messages. For spam, SVM achieved a precision of 0.99 and a recall of 0.90, resulting in a high F1-score of 0.94. This result indicates that SVM was particularly effective at capturing spam messages without a significant drop in precision, making it one of the best models for this task. Logistic Regression also performed well, achieving a test accuracy of 0.9803. The model classified ham messages with high precision, recall, and F1-score, all close to 0.98, 1.00, and 0.99, respectively. For spam, Logistic Regression achieved a precision of 0.99 and a recall of 0.86, resulting in an F1-score of 0.92. While its performance was slightly lower than SVM, it remained competitive, especially in terms of precision.

The Ensemble Voting Classifier, which combines the predictions of the individual models, achieved a test accuracy of 0.9848. For ham, the precision, recall, and F1-score were 0.98, 1.00, and 0.99, respectively. The performance on spam was notably strong, with precision at 0.99, recall at 0.89, and an F1-score of 0.94. The ensemble approach effectively balanced the strengths of the individual models, resulting in high overall accuracy and improved recall for spam messages. This suggests that combining models allows for better performance across diverse types of messages, making the ensemble a robust choice for spam detection. Analyzing the results further, the Support Vector Machine and the Ensemble Voting Classifier demonstrated the highest performance, with accuracies of 0.9857 and 0.9848, respectively. Both models exhibited a balance between precision and recall for both ham and spam messages, reducing the number of false negatives for spam detection, a critical factor in any spam classification system. These models' superior recall for spam compared to the other models is particularly noteworthy, as it indicates their effectiveness in identifying spam messages without sacrificing precision.

Discussions

While Random Forest, Gradient Boosting, and AdaBoost performed well in terms of overall accuracy, their slightly lower recall for spam indicates that they may miss some spam messages, misclassifying them as legitimate. This is a known challenge in imbalanced datasets, where the majority class (ham) can dominate the learning process. Despite this, these models still provided robust results and would be useful in situations where precision is more critical than

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

recall. Feature importance analysis conducted using the Random Forest model provided insights into the terms most associated with spam and ham messages. Common spam-related terms included promotional words, free offers, and specific numbers or symbols often associated with scams or contests. This behavior aligns with the expected characteristics of spam messages, highlighting the model's ability to capture key features of the spam class effectively.

The experimental results show that machine learning models, particularly Support Vector Machine and ensemble methods, are highly effective for detecting SMS spam. However, future improvements could focus on addressing the imbalance between ham and spam messages by employing more advanced resampling techniques, such as Synthetic Minority Over-sampling Technique (SMOTE), or exploring alternative ensemble strategies that further enhance recall for spam messages. Additionally, exploring more sophisticated models, such as deep learning approaches, might lead to further improvements, particularly in capturing more subtle spam patterns that are difficult for traditional machine learning models to detect. In conclusion, while all models performed well, Support Vector Machine and the Ensemble Voting Classifier emerged as the most effective for SMS spam detection, particularly in maintaining a balance between precision and recall. These models offer a robust solution for real-world spam detection systems, where minimizing false negatives is crucial for maintaining system effectiveness and user trust.

Table 1The results of machine learning models

Model	Accuracy	Precision (ham)	Recall (ham)	F1-score (ham)	Precision (spam)	Recall (spam)	F1-score (spam)
Random Forest	0.974	0.97	1.0	0.99	1.0	0.81	0.89
Gradient Boosting	0.9713	0.97	0.99	0.98	0.96	0.82	0.88
AdaBoost	0.9713	0.97	0.99	0.98	0.95	0.83	0.88
Support Vector Machine	0.9857	0.98	1.0	0.99	0.99	0.9	0.94
Logistic Regression	0.9803	0.98	1.0	0.99	0.99	0.86	0.92
Ensemble Voting Classifier	0.9848	0.98	1.0	0.99	0.99	0.89	0.94

CONCLUSION

This study evaluates the performance of various machine learning models for SMS spam detection, including Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, and an Ensemble Voting Classifier. In response to the growing need for accurate spam detection due to the rise of unsolicited messages, the research leverages hyperparameter tuning to compare these models. SVM and the Ensemble Voting Classifier were the most effective, with SVM achieving the highest test accuracy (0.9857) and near-perfect precision and recall for both ham and spam messages, while the Ensemble Voting Classifier, with an accuracy of 0.9848, offered improved recall for spam, reducing false negatives. Although Random Forest, Gradient Boosting, and AdaBoost performed well, they showed lower recall for spam, suggesting they might be more appropriate when precision is prioritized over recall. Feature importance analysis highlighted the models' ability to identify spam-related terms, but the class imbalance between ham and spam posed challenges, potentially influencing model performance. Addressing this imbalance through resampling techniques could enhance overall performance. Machine learning models, particularly SVM and ensemble methods, demonstrate robust solutions for spam detection, offering high precision and recall suitable for real-world applications where minimizing false negatives is critical. Future work should explore class imbalance solutions and investigate deep learning models for further improvements.

REFERENCES

Abid, M. A., Ullah, S., Siddique, M. A., Mushtaq, M. F., Aljedaani, W. & Rustam, F. (2022). Spam SMS filtering based on text features and supervised machine learning techniques. *Multimedia Tools and Applications, 81*(28), 39853–39871.

Abu-Salih, B., Qudah, D. Al, Al-Hassan, M., Ghafari, S. M., Issa, T., Aljarah, I., Beheshti, A. & Alqahtani, S. (2022). An intelligent system for multi-topic social spam detection in microblogging. *Journal of Information Science*, 01655515221124062.

Afifi, S., GholamHosseini, H. & Sinha, R. (2020). FPGA implementations of SVM classifiers: A review. *SN Computer Science, 1*(3), 133.

Alam, T. M., Shaukat, K., Khan, W. A., Hameed, I. A., Almuqren, L. A., Raza, M. A., Aslam, M. & Luo, S. (2022).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

An efficient deep learning-based skin cancer classifier for an imbalanced dataset. *Diagnostics*, *12*(9), 2115.

Alkhalil, Z., Hewage, C., Nawaf, L. & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, *3*, 563060.

Awe, O. O., Opateye, G. O., Johnson, C. A. G., Tayo, O. T. & Dias, R. (2024). Weighted hard and soft voting ensemble machine learning classifiers: Application to anaemia diagnosis. In *Sustainable Statistical and Data Science Methods and Practices: Reports from LISA 2020 Global Network, Ghana, 2022* (pp. 351–374). Springer.

Bose, S. (2023). *Deep One-Class Learning for Anomalous Short-text Classification*.

Božanić, M. & Sinha, S. (2021). *Mobile communication networks: 5G and a vision of 6G*. Springer.

Carmona, P., Dwekat, A. & Mardawi, Z. (2022). No more black boxes! Explaining the predictions of a machine learning XGBoost classifier algorithm in business failure. *Research in International Business and Finance*, *61*, 101649.

Choi, J., Jeon, B. & Jeon, C. (2024). Scalable Learning Framework for Detecting New Types of Twitter Spam with Misuse and Anomaly Detection. *Sensors*, *24*(7), 2263.

Daisy, S. J. S. & Begum, A. R. (2021). Smart material to build mail spam filtering technique using Naive Bayes and MRF methodologies. *Materials Today: Proceedings*, *47*, 446–452.

Dapat, V. (2024). *SMS Spam Detection Dataset*. https://www.kaggle.com/datasets/vishakhdapat/sms-spam-detection-dataset/data

de Zarzà, I., de Curtò, J. & Calafate, C. T. (2023). Optimizing Neural Networks for Imbalanced Data. *Electronics*, *12*(12), 2674.

Dhieb, N., Ghazzai, H., Besbes, H. & Massoud, Y. (2020). A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement. *IEEE Access*, *8*, 58546–58558.

Fayaz, M., Khan, A., Rahman, J. U., Alharbi, A., Uddin, M. I. & Alouffi, B. (2020). Ensemble machine learning model for classification of spam product reviews. *Complexity*, *2020*(1), 8857570.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M. & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, *115*, 105151.

Gaye, B., Zhang, D. & Wulamu, A. (2021). Improvement of support vector machine algorithm in big data background. *Mathematical Problems in Engineering*, *2021*(1), 5594899.

Genuer, R., Poggi, J.-M., Genuer, R. & Poggi, J.-M. (2020). *Random forests*. Springer.

Islam, M. K., Al Amin, M., Islam, M. R., Mahbub, M. N. I., Showrov, M. I. H. & Kaushal, C. (2021). Spam-detection with comparative analysis and spamming words extractions. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, 1–9.

Jáñez-Martino, F., Alaiz-Rodr\'\iguez, R., González-Castro, V., Fidalgo, E. & Alegre, E. (2023). A review of spam email detection: analysis of spammer strategies and the dataset shift problem. *Artificial Intelligence Review*, *56*(2), 1145–1173.

Kulkarni, A., Balachandran, V. & Das, T. (2024). Phishing Webpage Detection: Unveiling the Threat Landscape and Investigating Detection Techniques. *IEEE Communications Surveys \& Tutorials*.

Le Jeune, L., Goedeme, T. & Mentens, N. (2021). Machine learning for misuse-based network intrusion detection: overview, unified evaluation and feature choice comparison framework. *Ieee Access*, *9*, 63995–64015.

Ling, R., Fortunati, L., Goggin, G., Lim, S. S. & Li, Y. (2020). *The Oxford handbook of mobile communication and society*. Oxford University Press.

Maqsood, U., Ur Rehman, S., Ali, T., Mahmood, K., Alsaedi, T. & Kundi, M. (2023). An Intelligent Framework Based on Deep Learning for SMS and e-mail Spam Detection. *Applied Computational Intelligence and Soft Computing*, *2023*(1), 6648970.

Maurya, S. K., Singh, D. & Maurya, A. K. (2023). Deceptive opinion spam detection approaches: a literature survey. *Applied Intelligence*, *53*(2), 2189–2234.

Mienye, I. D. & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, *10*, 99129–99149.

Mushtaq, Z., Ramzan, M. F., Ali, S., Baseer, S., Samad, A. & Husnain, M. (2022). Voting Classification-Based Diabetes Mellitus Prediction Using Hypertuned Machine-Learning Techniques. *Mobile Information Systems*, *2022*(1), 6521532.

Noekhah, S., binti Salim, N. & Zakaria, N. H. (2020). Opinion spam detection: Using multi-iterative graph-based model. *Information Processing \& Management*, *57*(1), 102140.

Patil, L., Sakhidas, J., Jain, D., Darji, S. & Borhade, K. (2022). A Comparative Study of Spam SMS Detection Techniques for English Content Using Supervised Machine Learning Algorithms. *International Symposium on*

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 4, October 2024
https://doi.org/10.47709/cnahpc.v6i4.4822

Submitted : Oct 10, 2024
Accepted : Oct 22, 2024
Published : Oct 31, 2024

Intelligent Informatics, 211–224.
Prosise, J. (2022). *Applied machine learning and AI for engineers*. " O'Reilly Media, Inc."
Rao, S., Verma, A. K. & Bhatia, T. (2021). A review on social spam detection: Challenges, open issues, and future directions. *Expert Systems with Applications*, *186*, 115742.
Rida, J. F. A. (2021). Overview of Development performance for Mobile Phone Wireless Communication Networks. *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 1–11.
Roy, P. K., Singh, J. P. & Banerjee, S. (2020). Deep learning to filter SMS Spam. *Future Generation Computer Systems*, *102*, 524–533.
Saidani, N., Adi, K. & Allili, M. S. (2020). A semantic-based classification approach for an enhanced spam detection. *Computers \& Security*, *94*, 101716.
Shaaban, M. A., Hassan, Y. F. & Guirguis, S. K. (2022). Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. *Complex \& Intelligent Systems*, *8*(6), 4897–4909.
Sharaff, A., Kamal, C., Porwal, S., Bhatia, S., Kaur, K. & Hassan, M. M. (2021). Spam message detection using Danger theory and Krill herd optimization. *Computer Networks*, *199*, 108453.
Swarnkar, M., Sharma, N. & Kumar Thakkar, H. (2022). Malicious URL detection using machine learning. In *Predictive Data Security using AI: Insights and Issues of Blockchain, IoT, and DevOps* (pp. 199–216). Springer.
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*.
Weichbroth Pawełand Łysik, Ł. (2020). Mobile security: Threats and best practices. *Mobile Information Systems*, *2020*(1), 8828078.
Zhang, H., Quost, B. & Masson, M.-H. (2023). Cautious weighted random forests. *Expert Systems with Applications*, *213*, 118883.
Zhou, X., Lu, P., Zheng, Z., Tolliver, D. & Keramati, A. (2020). Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree. *Reliability Engineering \& System Safety*, *200*, 106931.

* Corresponding author