Jurnal Artificial, Vol 4 No 2, Oktober 2026

Implementation of Naive Bayes and Support Vector Machine for SMS Spam Classification Using the SMS Spam Collection Dataset

Muhammad Abrar Rayhan [1*]

[1] Telkom University, Indonesia

* abrarrayhan8@gmail.com

Abstract

Short Message Service (SMS) is one of the most popular communication services on mobile networks. The rapid proliferation of mobile communication has led to an increase in spam messages offering ads, false links, and misinformation, which could pose a threat to user privacy. Automated spam detection using machine learning methods has become a key approach to tackling this problem in recent years. The aim of this research is to apply and train on how the SMS Spam Collection. Dataset for SMS spam classification using 2 machine learning algorithms, Naïve Bayes and Support Vector Machine (SVM). Several steps are taken, including data preprocessing, text cleanup, feature extraction using the Term Frequency–Inverse Document Frequency (TF-IDF) method, and model training. The performance of the implemented models is assessed using accuracy, precision, recall, F1-score, a confusion matrix, and cross-validation. The results from the experiments show that both algorithms can successfully classify these SMS spam messages. However, the Support Vector Machine model outperforms the Naïve Bayes model, achieving an accuracy of nearly 98% on the classification task. These results demonstrate that machine learning techniques, including Support Vector Machine in combination with TF-IDF feature extraction, provide reliable performance for SMS spam detection, and could be helpful for an automated filter system in m-commerce services.

Keywords: *SMS Spam Detection, Machine Learning, Text Classification, Naïve Bayes, Support Vector Machine*

Introduction

Short Message Service (SMS) remains one of the most widely used communication services due to its simplicity, low cost, and reliability. Despite the rapid development of internet-based messaging platforms, SMS continues to play a significant role in communication, particularly for notifications, authentication systems, and business messaging. However, the increasing reliance on SMS has also led to a significant rise in spam messages, including unsolicited advertisements, phishing attempts, and fraudulent content. These spam messages not only reduce communication quality but also pose serious threats to user privacy and security.

Recent studies and industry reports indicate that spam messages still account for a substantial proportion of global mobile traffic. The evolution of spam techniques, including the use of obfuscated text, abbreviations, and dynamic message patterns, has made traditional rule-based filtering methods increasingly ineffective. As a result, there is a growing need for intelligent and adaptive approaches that can automatically detect and classify spam messages with high accuracy.

To address this challenge, machine learning (ML) techniques have been widely adopted in SMS spam detection. Various algorithms, including Naïve Bayes (NB), Support Vector Machine (SVM),

https://doi.org/10.54065/artificial.1249

Decision Tree, and Logistic Regression, have been applied to classify SMS messages into spam and legitimate categories. Among these, NB is known for its simplicity and efficiency in probabilistic text classification, while SVM is particularly effective in handling high-dimensional data and complex decision boundaries.

In addition to classification algorithms, feature extraction plays a crucial role in improving model performance. Since textual data cannot be directly processed by ML models, it must first be transformed into numerical representations. One of the most widely used techniques is Term Frequency–Inverse Document Frequency (TF-IDF), which captures the importance of words in a document relative to a corpus. TF-IDF has been extensively used in text mining and information retrieval due to its effectiveness in distinguishing relevant and irrelevant terms.

Previous studies have demonstrated that machine learning models can achieve promising results in SMS spam classification. For instance, conducted a comparative analysis of several classification algorithms and found that SVM often outperforms other models in high-dimensional text classification tasks (Pranckevičius et al, 2017). Similarly, other studies have reported strong performance of NB due to its efficiency and robustness in handling text-based data. However, the reported results across studies vary significantly depending on the dataset, preprocessing techniques, feature extraction methods, and evaluation metrics used.

Despite these advancements, several limitations remain. Many previous studies employ different experimental settings, making it difficult to conduct fair and reproducible comparisons between algorithms. In addition, variations in dataset characteristics and preprocessing pipelines can significantly influence model performance. As a result, it is still unclear how different machine learning algorithms perform under a consistent and controlled experimental framework.

Furthermore, most studies focus on achieving high accuracy without providing a comprehensive evaluation using multiple performance metrics. In classification problems such as spam detection, relying solely on accuracy may lead to misleading conclusions, especially in imbalanced datasets. Therefore, it is important to evaluate models using additional metrics such as precision, recall, F1-score, confusion matrix, and cross-validation to obtain a more complete understanding of model performance.

Based on these challenges, this study aims to provide a systematic evaluation of two widely used machine learning algorithms, Naïve Bayes and Support Vector Machine, for SMS spam classification. The models are implemented using a unified preprocessing and feature extraction pipeline to ensure fair comparison. The dataset used in this study is the SMS Spam Collection dataset, which consists of 5,574 labeled SMS messages, including 4,827 legitimate (ham) messages and 747 spam messages.

The methodology includes data preprocessing, text cleaning, feature extraction using TF-IDF, and model training. The performance of the models is evaluated using multiple metrics, including accuracy, precision, recall, F1-score, confusion matrix, and cross-validation. By applying a consistent experimental setup, this study aims to provide reliable and reproducible results for comparing the effectiveness of NB and SVM in SMS spam classification.

The main contribution of this study is to present a structured and comprehensive comparison of machine learning algorithms under a unified framework. This research provides insights into the strengths and limitations of each algorithm and highlights the importance of consistent evaluation in text classification tasks. The findings of this study are expected to contribute to the development of more accurate and reliable automated SMS spam detection systems.

The remainder of this paper is organized as follows. Section 2 describes the research methodology, including data preprocessing and model implementation. Section 3 presents the experimental results and analysis. Finally, Section 4 concludes the study and provides recommendations for future research.

Method

Research Design

This study employs a quantitative experimental research design using a machine learning approach to classify SMS messages into spam and non-spam categories. The overall workflow consists of dataset collection, data preprocessing, feature extraction, model development, and performance evaluation. The dataset used in this study is the SMS Spam Collection Dataset, a publicly available benchmark dataset for spam detection research (Abayomi-Alli et al, 2022). The dataset contains 5,574 SMS messages, consisting of 4,827 legitimate (ham) messages and 747 spam messages.

In the preprocessing stage, the text data is cleaned by removing punctuation, special characters, and irrelevant symbols. The text is also normalized through case folding and tokenization. Since machine learning algorithms require numerical input, the cleaned text is transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method. TF-IDF is used to measure the importance of words in a document relative to the entire dataset. Two classification algorithms are implemented in this study: Naïve Bayes (NB) and Support Vector Machine (SVM). These models are selected due to their effectiveness in handling high-dimensional textual data and their strong performance in text classification tasks (Zhang et al, 2017).

Model training and evaluation are conducted using Python programming language with libraries such as scikit-learn, pandas, and NumPy. The performance of the models is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score, to determine their effectiveness in detecting spam messages.

Dataset Description

In this study, the dataset used is the SMS Spam Collection Dataset, which is a publicly available dataset commonly used in SMS spam detection research. The dataset has 5,574 SMS messages including 4,827 legitimate messages (ham) and 747 spam messages. Initially, the dataset was gathered from the NUS SMS Corpus, Grumbletext website, and other public SMS collection sites. Thus, the SMS Spam Collection Dataset from Kaggle is the dataset used in this research. The dataset consists of 5,574 SMS messages which are divided into two classes: Spam and Ham (natural text).

There is a statement written to the data entry that has only two attributes: the message label and the message. As this dataset is fairly evenly divided between spam and non-spam messages, it is used extensively as the benchmark in spam classification research. Each SMS message in the dataset contains a label indicating whether it is spam or ham, along with the message text. The dataset provides a realistic representation of SMS communication and is commonly used in machine learning experiments for text classification tasks.

Data Preprocessing

Text data preprocessing — preprocessing text data helps clean and prepare the dataset for training machine learning models. Texts in raw SMS messaging systems are often messy, with factors such as punctuation, URLs, and text formatting that may hinder classification. Thus, some preprocessing methods are used to enhance the quality of the dataset. For the first step, all SMS messages are converted to lowercase to standardize the

text and avoid duplicate words from different cases. Punctuation marks, special characters, and URLs are also removed, as they do not contribute to the classification process. Character normalization removes irrelevant symbols that may be produced in SMS messages. Subsequently, tokenization is performed to break each SMS message into individual words.

This stage helps the model handle textual data at the word level. Stopword editing is then used to remove widely used words like "the", "is", "and", "to", which, in general, do not serve an informative role in classification problems. Furthermore, stemming is an alternative for reverting words to their original forms to save language space and computational resources. Words including "winning", "winner", and "wins" are replaced by the root "win". Finally, the duplicate SMS messages are eliminated to avoid bias in model training and evaluation. These preprocessing steps reduce the noise in the dataset and optimise the performance of the machine learning models used in spam classification.

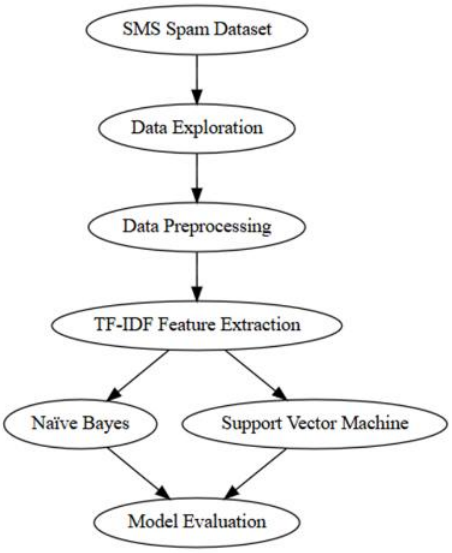

Figure 1. *Research Methodology Pipeline for SMS Spam Classification*

Figure 1. Research Methodology Pipeline for SMS Spam Classification. The figure illustrates the sequential stages of the study, beginning with data collection, followed by preprocessing, feature extraction, model training, classification, and evaluation to identify spam and legitimate SMS messages accurately.

Feature Extraction

Feature extraction converts text into numerical representations for machine learning algorithms. Text data is converted to numerical feature vectors with TF-IDF. This is a popular text-based processing technique for creating numerical feature vectors on documents and other similar types of documents. TF-IDF is a general technique commonly used in text mining and natural language processing and also quantifies the importance of a word in a document relative to a whole dataset. The Term Frequency (TF) indicates the frequency of a word in a document, whereas the Inverse Document Frequency (IDF) expresses how unique or informative the word is across all documents.

Because TF-IDF uses the combination of these two measures, words with a high frequency in one document but rarely appear in other documents, therefore, they get a higher weight. This representation in its output allows machine learning algorithms to

discover key textual structures and enhances classification results for text mining applications.TF-IDF feature representation has been widely used in spam detection and text classification research due to its effectiveness in handling high-dimensional textual data (Raschka, 2018; Saeed, 2023).

Model Implementation

In this study, two machine learning algorithms are implemented for SMS spam classification: Naïve Bayes and Support Vector Machine (SVM). Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem, which assumes independence among features. Due to its simplicity and computational efficiency, Naïve Bayes has been widely used in text classification tasks, including spam detection (Wafda, 2024; Saeed, 2023).

The Support Vector Machine (SVM) is a supervised machine learning technique in which the goal is to create an optimal hyperplane that separates data points belonging to different classes in a high-dimensional feature space. SVM performs well in text classification problems due to its effective treatment of high-dimensional feature vectors formed from text data. Here, we divide the dataset into a training set and a testing set. The data split is 80:20, with 80% for model training and 20% for testing. The classification model is created based on the training process and then evaluated based on the testing process to ensure that new data is tested independently.

Machine Learning Models

In this study, two machine learning algorithms are implemented to perform SMS spam classification: Naïve Bayes and Support Vector Machine (SVM). These algorithms are widely used in text classification tasks due to their effectiveness in handling high-dimensional textual data. Both algorithms are trained using feature vectors generated from the TF-IDF feature extraction process. The use of machine learning algorithms enables automatic classification of SMS messages into spam and legitimate categories by learning patterns from labeled datasets. Previous studies have demonstrated that these algorithms can achieve strong performance in spam detection and text classification problems (Wafda, 2025; Johari et al., 2025).

Naïve Bayes

Naïve Bayes is a Bayes' theorem based probabilistic classification algorithm that determines the posterior probability of a class given the observed features. Under the class label conditionality, each feature is independent of all others. While this assumption does not always hold in real-world datasets, Naïve Bayes does well on several text classification tasks thanks to its simplicity and computational efficiency. For this reason, the Multinomial Naïve Bayes variant is widely used in text mining applications, especially for modeling discrete features (e.g., word frequencies) or TF-IDF representations extracted from textual data. Previous studies have shown that Naïve Bayes performs efficiently in text classification tasks including sentiment analysis and review classification (Dey et al., 2021).

$$P(C|X) = (P(X|C)P(C))/(P(X))$$

Where P(C|X) is the posterior probability of class C given the feature vector X, P(X|C) is the likelihood of observing feature X given class C, P(C) denotes the prior probability of class C, and P(X) denotes the evidence probability. For text classification tasks, the

Multinomial Naïve Bayes variant is commonly used because it is well suited for modeling discrete features such as word frequencies or TF-IDF representations extracted from textual data. Previous studies have shown that Naïve Bayes performs efficiently in text classification tasks including sentiment analysis and review classification (Dey et al., 2021).

Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification and regression. The main objective of SVM is to find an optimal hyperplane that separates data points belonging to different classes in a high-dimensional feature space. The optimal hyperplane is determined by maximizing the margin between the closest data points of different classes, known as support vectors. SVM is particularly effective in text classification tasks because textual data usually produces high-dimensional feature vectors after applying feature extraction techniques such as TF-IDF.

According to researcher, SVM provides strong generalization capability and can effectively classify data by constructing optimal decision boundaries between classes (Anggraini et al, 2024). Several studies have demonstrated the effectiveness of SVM in classification tasks involving textual datasets and other machine learning applications (Krishnaveni et al, 2021). In this research, a Kernel is used in the SVM model because it is commonly applied in text classification tasks and provides efficient performance when dealing with large textual feature spaces.

Mathematically, the decision boundary of SVM can be expressed as: $w \cdot x+b=0$

Where W represents the weight vector, X Represents the feature vector, and b represents the bias term. The goal of SVM is to determine the optimal values of W and B that maximize the margin between different classes. Text classification operations are best suited for SVM due to high-dimensional feature vectors arising from textual data when feature extraction techniques (including TF-IDF) are applied. According to researcher, SVM provides strong generalization capability and can effectively classify data by constructing optimal decision boundaries between classes (Gupta et al, 2021).

Several studies have demonstrated the effectiveness of SVM in classification tasks involving textual datasets and other machine learning applications (Setiyono et al, 2019). In this research, a linear kernel is used in the SVM model because it is commonly applied in text classification tasks and provides efficient performance when dealing with large textual feature spaces.

Experimental Setup

Within the experiment's framework, the dataset can be split into a training and a test set. The classification models are trained on the training data, and evaluation is performed on the test data. The dataset in this investigation is divided proportionally into 80% for training and 20% for testing. This provides a better estimate of the model's accuracy on unseen data, as the models are evaluated on data (not the observations themselves).

Model Evaluation Metrics

To evaluate the performance of the classification models, this study uses several metrics, including accuracy, precision, recall, and F1-score. These metrics are widely used in machine learning classification tasks to evaluate a model's ability to correctly predict whether a message is spam or non-spam. In addition to these metrics, a confusion matrix

is used to provide a detailed evaluation of the classification results. A confusion matrix shows the number of correctly and incorrectly classified instances generated by the model. This evaluation method is widely used in classification problems because it provides a clearer understanding of prediction performance beyond overall accuracy (Singh et al, 2021).

The confusion matrix consists of four possible outcomes: 1). True Positive (TP) represents correctly classified spam messages; 2). True Negative (TN) represents legitimate messages correctly classified as ham; 3). False Positive (FP) represents legitimate messages incorrectly classified as spam; 4). False Negative (FN) represents spam messages incorrectly classified as legitimate messages. Based on these values, several evaluation metrics can be calculated to assess the performance of the classification models.

1. Accuracy: Accuracy measures the proportion of correctly classified messages among the total number of messages.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision Precision measures the proportion of correctly predicted spam messages among all messages predicted as spam:

$$Precision = \frac{TP}{TP + FP}$$

3. Recall: Recall measures the proportion of actual spam messages that are correctly identified by the model:

$$Recall = \frac{TP}{TP + FN}$$

4. F1-score: The F1-score represents the harmonic mean of precision and recall and provides a balanced evaluation of the classification model.

$$F1 = 2\ X\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

These evaluation metrics provide a comprehensive assessment of the classification performance and help determine the effectiveness of the machine learning models in detecting SMS spam messages.

Cross Validation

In this study, cross-validation is used to evaluate the reliability and robustness of the classification models. Cross-validation divides the dataset into several subsets and performs multiple rounds of training and testing. In this research, k-fold cross-validation is applied to assess the consistency of model performance across different data splits. In k-fold cross-validation, the dataset is divided into k equal subsets. During each iteration, one subset is used as the testing data while the remaining subsets are used as training data. This process is repeated k times, allowing each subset to be used once as testing data. This approach helps reduce the risk of overfitting and provides a more reliable estimation of the model's generalization performance.

Results

In the present chapter, we describe and elaborate on experiment results generated from the applying of the Naïve Bayes and Support Vector Machine algorithms for SMS spam classification. All experiments were conducted on the SMS Spam Collection Dataset with TF-IDF feature extraction. Several indices to evaluate the performance of the models are accuracy, precision, recall, F1-score, confusion matrix, and cross-validation. The results are then analyzed to assess the performance of each of the algorithms in classifying spam and legitimate SMS messages.

Dataset Analysis

The SMS Spam Collection Dataset used in this study contains 5,572 SMS messages, labeled as spam or ham. Understanding the dataset's distribution is important because an imbalanced dataset can affect the performance of machine learning models.

Table 1. *Distribution of SMS Messages in the Dataset*

Label	Number of Messages
Ham	4516
Spam	653
Total	5572

The dataset shows that the number of legitimate (ham) messages is significantly larger than the number of spam messages. This distribution reflects real-world SMS communication, where most messages are legitimate while only a small portion is spam. Such an imbalance may affect classification performance; therefore, appropriate evaluation metrics are required to assess model performance.

Wordcloud Analysis

To gain more insights into spam message classification in the dataset, a WordCloud is generated to show the most common words in spam SMS messages. WordCloud is a commonly used visualization technique in text mining that highlights the relative frequency of words within a corpus, where larger words indicate higher occurrence in the dataset (Shahbazov, 2026). The WordCloud created from the spam messages in the SMS Spam Collection dataset is shown in Figure 2. The visualization shows how frequently terms such as free, call, text, claim, reply, and prize appear.

These phrases are typically used in spam messages to promote or persuade users, and to get their attention. Similar findings have been reported in previous studies on SMS spam detection, where spam messages frequently contain promotional keywords designed to manipulate user behavior (Giri et al, 2023). WordCloud visualization enables users to intuitively interpret the features of spam messages and supports TF-IDF feature extraction, where informative words are given higher weights in spam messages.

Figure 2. *WordCloud visualization of spam messages*

Figure 2. WordCloud Visualization of Spam Messages. The figure presents the most frequently occurring words in spam messages, where larger and more prominent words indicate higher frequency, highlighting common terms often used in spam communication.

Model Performance

After preprocessing and feature extraction using TF-IDF, the classification models were trained using Naïve Bayes and Support Vector Machine algorithms. The performance of both models was evaluated using several metrics including accuracy, precision, recall, and F1-score. The experimental results of the two models are presented in Table 2.

Table 2. *Performance Comparison of Naïve Bayes and SVM Models*

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.957447	1.000	0.696552	0.821138
SVM	0.978723	0.992	0.855172	0.918519

The results indicate that both machine learning models can effectively classify SMS spam messages. However, the Support Vector Machine model achieved better performance than Naïve Bayes.

Confusion Matrix Analysis

We apply a confusion matrix to assess how well our implemented models can classify SMS messages into spam and legitimate categories. In the confusion matrix, we get a more detailed picture of how many true positives, true negatives, false positives, and false negatives our classification model produced. This evaluation method is commonly used in machine learning classification tasks because it provides a clearer understanding of the model's prediction behavior beyond overall accuracy (Ugbotu et al, 2025). The confusion matrix using the Support Vector Machine model is presented in Figure 1. The matrix shows how many spam messages were correctly classified as spam and how many legitimate messages were correctly classified as ham. In addition, the confusion matrix provides the number of misclassified messages, allowing the types of classification errors contributing to the model to be inferred.

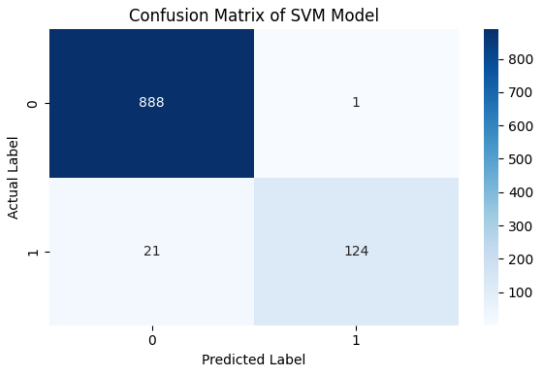

Figure 3. *Confusion Matrix of the Support Vector Machine (SVM) Model*

On the confusion matrix, it clearly indicates that most SMS messages are detected correctly by the model. Based on the confusion matrix, the model correctly classified 888 legitimate messages and 124 spam messages. Only 1 legitimate message was incorrectly predicted as spam, while 21 spam messages were misclassified as legitimate messages. These results indicate that the model performs well in identifying legitimate messages and detecting spam messages with only a small number of misclassification errors.

ROC Curve Analysis

Receiver Operating Characteristic (ROC) analysis is done to evaluate the performance of the implemented models over various decision thresholds. The ROC curve is widely used in machine learning classification tasks because it provides a graphical representation of the trade-off between the true positive rate and the false positive rate (Roy et al, 2020). A curve that approaches the top-left corner of the graph indicates better performance in classification. Here, AUC is usually utilized to assess the overall effectiveness of the model in distinguishing between the two classes.

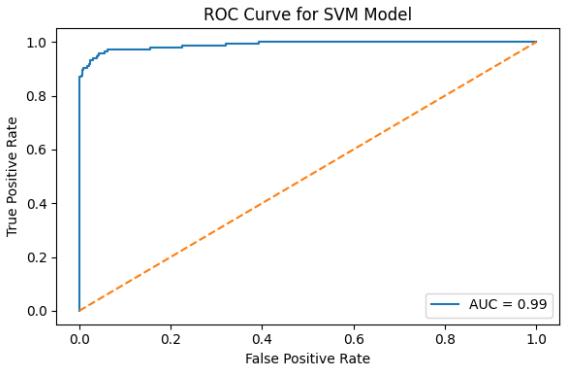

***Figure 4**. ROC Curve of the Support Vector Machine (SVM) Model*

From the ROC curve shown in Figure 2, the Support Vector Machine performs well at classifying information. The AUC value derived from the experiment is nearly 1, indicating strong performance in distinguishing spam from normal messages. An AUC value close to 1 indicates that the model successfully ranks spam messages higher than non-spam messages. The results verify the reliable performance of TF-IDF feature extraction and Support Vector Machine classification for SMS spam detection.

Consequently, this approach can be an effective means of providing automatic spam filtering in mobile communication service applications. The strong performance of the SVM model in ROC analysis is consistent with previous studies which indicate that Support Vector Machine is highly effective for text classification tasks involving high-dimensional feature spaces (Nagwani et al, 2017). Furthermore, the use of TF-IDF feature extraction also contributes to improving classification performance by capturing the importance of words within SMS messages (Johari et al., 2025).

Cross Validation Result

BTo test the robustness and stability of the implemented machine learning model, cross-validation was performed in this study. Cross-validation is one of the key evaluation tools in machine learning to analyze how well a model generalizes to new data. The technique splits the data into multiple subsets, which are used in a repeating way as training data and as testing data. Cross-validation makes model performance estimation more reliable through multiple iterations of training and testing.

The current study performed a 5-fold cross-validation on the Support Vector Machine model. The data had five subsets, and the model was trained and evaluated 5 times with diverse training/test datasets. This approach helps reduce the risk of overfitting, guaranteeing model performance consistency over varying data partitions. Results of the cross-validation experiment are shown in Table 3.

Table 3. *Cross-Validation Accuracy Results of the SVM Model*

Fold	Accuracy
1	0.979691
2	0.974855
3	0.973888
4	0.973888
5	0.971926
Mean Accuracy	0.974850

The cross-validation outputs of the Support Vector Machine model are shown in Table 3. Accuracy values across the five folds range from 0.972 to 0.980. The average accuracy of this experiment shows that the model performs consistently across data partitions. Thus, the results show that the Support Vector Machine model has good generalization capability and is not significantly affected by variations in the training dataset. In this regard, the model may be considered reliable for SMS spam classification tasks (Wafda, 2024).

Discussion

These experimental results confirm the effectiveness of machine learning algorithms in classifying SMS messages into spam and non-spam categories. In particular, the Support Vector Machine (SVM) model outperformed the Naïve Bayes model across multiple evaluation metrics (accuracy, recall, and F1-score). It can be seen that SVM achieves more accurate classification with TF-IDF for SMS spam. This supports the pattern of textual data, which is typically high-dimensional and sparse. In such cases, SVM is well established as a successful algorithm because it can generate a suitable decision boundary that effectively separates classes in a high-dimensional feature space. The performance comparison of Naïve Bayes and SVM indicates that, despite their ability for text classification tasks, SVM generalizes better.

Naïve Bayes makes probabilistic assumptions and requires conditional independence, which may not hold for real-world text data. In contrast, SVM is focused on maximizing the margin between classes and is therefore better suited to dealing with complex relationships between textual representations of features. Therefore, SVMs can achieve better classification accuracy in high-dimensional feature spaces and in spaces with feature-feature-value pairs that are not independent. The confusion matrix provides additional analysis of the classification outcomes. The confusion matrix contains the number of correctly classified and incorrectly classified messages per class. According to the confusion matrix, the model correctly classified 888 legitimate (ham) messages and 124 spam messages. This led to only a handful of misclassified messages one was a legitimate message incorrectly categorized as spam, and 21 others were misclassified as legitimate communications. This means the classification model can separate spam from valid messages with relatively high accuracy, resulting in few classification errors.

The low proportions of false positives and false negatives also indicate that the classification model works well. In spam detection systems, false positives are genuine messages incorrectly classified as spam. This kind of mistake is especially horrible because it could mean someone blocks an essential message or renders it completely irrelevant. The results of our research reveal that false positives remain almost entirely low, demonstrating that the model can retain legitimate messages while still efficiently identifying spam.

On the contrary, the number of false negatives is relatively small, indicating that the model can efficiently detect most spam messages. Apart from confusion matrix analysis, ROC curve analysis also depicts the performance of the classification model. A ROC curve shows the relationship between true positive rate and false positive rate at different class thresholds. In this scenario, the SVM model's ROC curve reaches the upper-left corner of the graph, indicating a high classification rate.

The AUC value of the experiment is about 0.99, demonstrating good classification of spam messages from legitimate ones. If the AUC value is close to one, the model is very good at ranking spam messages above non-spam messages. The cross-validation data also confirm the reliability of the classification models in this experiment. A five-fold cross-validation experiment yields accurate performance across folds of the dataset, with a mean accuracy of approximately. 0.974850.

The slight difference in the folds indicates that the model consistently provides predictions regardless of the data partition. This result indicates that the classification model is not strongly dependent on a particular data split and achieves acceptable generalization performance on unseen data. The results of this study are in agreement with previous studies in text classification and spam detection. Prior research has demonstrated that Support Vector Machines can be applied more efficiently to multi-dimensional textual data than Naïve Bayes.

This is mainly because in complex feature spaces produced by TF-IDF-like feature-extraction algorithms, SVM can efficiently generate separating hyperplanes. Also, TF-IDF feature extraction is important for improving classification and assigning weights to informative words that are used in spam messages but are rarer in legitimate ones. Textual features also play a key role in spam message classification, as another observation arising from the experimental findings. From this study, we can see by the WordCloud visualization that most spam messages contain promotional or persuasive terms such as free, call, claim, prize, reply, etc.

The aforementioned words are often used in spam-like messages to pique people's interest and urge them to respond. The presence of such keywords helps the TF-IDF feature extraction distinguish spam from legitimate messages. The results of this paper suggest that combining TF-IDF with different machine learning algorithms (including Naïve Bayes and Support Vector Machine) is promising for SMS spam classification. By far, SVM achieved the best overall performance among our models, making it ideal for working with high-dimensional textual data. Our results demonstrate the applicability of an ML model to enable automated spam filtering in mobile communication scenarios.

Applying such techniques to tools enables a service provider to improve the accuracy of spam filtering systems and protect users from unintended or potentially harmful messages. In addition to the quantitative evaluation results, it is also important to consider the practical implications of implementing machine learning models for SMS spam detection systems. In real-world mobile communication environments, spam messages are continuously evolving in both structure and linguistic patterns. Spammers often modify message content to bypass existing filtering mechanisms by introducing variations in words, symbols, or formatting. Therefore, spam detection systems must rely on robust feature extraction methods and adaptable machine learning models that can generalize well across different forms of textual data.

The use of TF-IDF in this study helps capture the importance of words in the dataset, enabling the models to recognize patterns commonly associated with spam messages. Another aspect worth discussing is the role of data imbalance in spam detection tasks. As observed in the dataset distribution, the number of legitimate (ham) messages is significantly larger than the number of spam messages. Such imbalance can influence the performance of classification models because the algorithm may become biased toward the majority class. Despite this imbalance, the models used in this study were still able to achieve high classification accuracy and strong recall values for spam messages.

This indicates that the machine learning models are capable of learning meaningful patterns even when the dataset contains unequal class distributions. However, future research may explore additional techniques such as resampling methods, cost-sensitive learning, or ensemble models to further improve classification performance in imbalanced datasets. It is also important to acknowledge several limitations of this study. First, the experiments were conducted using a single dataset, namely the SMS Spam Collection Dataset. Although this dataset is widely used in spam detection research, it may not fully represent the diversity of SMS spam messages encountered in real-world scenarios. Spam messages may vary across regions, languages, and communication platforms.

Therefore, future studies could evaluate the proposed models on multiple datasets or multilingual SMS datasets to better assess the robustness of the classification approach. Second, this study focused on traditional machine learning algorithms, namely Naïve Bayes and Support Vector Machine. While these algorithms provide strong baseline performance, more advanced techniques such as deep learning models, including recurrent neural networks or transformer-based models, may provide further improvements in spam detection accuracy. Despite these limitations, the findings of this research contribute to the growing body of knowledge in spam detection and text classification. The results demonstrate that relatively simple machine learning models combined with appropriate feature extraction techniques can achieve high performance in SMS spam classification tasks. This highlights the potential of implementing lightweight and computationally efficient models for real-time spam filtering systems in mobile communication networks. Such systems can help reduce the spread of unwanted messages, protect users from fraudulent or malicious content, and improve the overall quality of digital communication services.

This study finding was also in agreement with other studies on SMS spam classification. Studies showed that Support Vector Machine usually outperforms Naïve Bayes in textual datasets where TF-IDF features have been used. Researcher found that SVM achieves good performance in high-dimensional text classification problems because of its capacity to generate optimal separating hyperplanes. These results are consistent with those produced after this study, where SVM performed higher accuracy and F1-score than Naïve Bayes.

Further studies on better feature extraction methods such as word embeddings, contextual language models, or deep learning models, could enhance detection of increasingly complex spam messages in the future. Also, investigating the proposed method on multilingual SMS datasets may better understand generalization of machine learning models over spam detection tasks. The results of this study have practical implications for the development of automated spam filtering systems in mobile communication services. By integrating machine learning models such as SVM with TF-IDF

feature extraction, mobile service providers can improve the accuracy of spam detection systems. This approach can help reduce the number of unwanted messages received by users and protect them from potential fraud, phishing attacks, or misleading promotional content.

Conclusion

This study aims to present a discussion of using the SMS Spam Collection dataset with a machine learning algorithm for SMS spam classification. Two classification methods, Naïve Bayes and Support Vector Machine (SVM), were provided and used to detect spam and non-spam messages. In this study, we can verify that both algorithms can effectively classify SMS packets using TF-IDF feature extraction at the experimental level. The SVM model performed better than Naïve Bayes for classification. The results of the evaluation show that SVM achieves higher accuracy and more reliable classification because of its ability to handle high-dimensional feature spaces derived from text documents. Also, the robustness of the models is not compromised in cross-validation, suggesting that the techniques perform consistently across folds and generalize quite well to new, uncharacterized data. However, this study is not without limitations. Firstly, we only ran on one dataset, which is the SMS Spam Collection dataset, and it could not be generalized widely to other spam messages, datasets, or data collection methods. Secondly, the approach does not investigate more modern algorithms such as deep learning, but rather conventional machine learning technologies so the lack of semantic understanding and limited generalization. For future research, more datasets should be analyzed, and deep learning models and transformer-based machine learning methods should also be examined, as well as other advanced machine learning approaches with a model-based architecture of deep learners. To optimize spam detection, future studies may also explore new features, such as semantic understanding or context-embedded approaches, to improve performance. On the whole, this study establishes that machine learning and TF-IDF feature-extraction-based approaches are capable of correctly recognizing spam messages and provide support for automatic spam filtering.

Acknowledgment

-

References

Abayomi-Alli, O., Misra, S., & Abayomi-Alli, A. (2022). A Deep Learning Method For Automatic SMS Spam Classification: Performance Of Learning Algorithms On Indigenous Dataset. *Concurrency And Computation: Practice And Experience*, *34*(17), E6989. Https://Doi.Org/10.1002/Cpe.6989

Dey, S., Wasif, S., Tonmoy, D. S., Sultana, S., Sarkar, J., & Dey, M. (2020). A Comparative Study Of Support Vector Machine And Naive Bayes Classifier For Sentiment Analysis On Amazon Product Reviews. In *2020 International Conference On Contemporary Computing And Applications (IC3A)* (Pp. 217-220). IEEE.

Johari, M. F., Chiew, K. L., Hosen, A. R., Yong, K. S., Khan, A. S., Abbasi, I. A., & Grzonka, D. (2025). Key Insights Into Recommended SMS Spam Detection Datasets. *Scientific Reports*, *15*(1), 8162. Https://Doi.Org/10.1038/S41598-025-92223-1

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison Of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, And Logistic Regression Classifiers For Text Reviews Classification. *Baltic Journal Of Modern Computing, 5*(2). Https://Doi.Org/10.22364/Bjmc.2017.5.2.05

Saeed, V. A. (2023). A Method For SMS Spam Message Detection Using Machine Learning. *Artif. Intell. Robot. Dev. J*, 3, 214-228. Https://Doi.Org/10.52098/Airdj.202366

Zhang, Y., & Wallace, B. (2017). A Sensitivity Analysis Of (And Practitioners' Guide To) Convolutional Neural Networks For Sentence Classification. *Journal Of Artificial Intelligence Research, 59*, 367–410. Https://Doi.Org/10.1613/Jair.5245

Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2019). Text Classification Algorithms: A Survey. *Information, 10*(4), 150. Https://Doi.Org/10.3390/Info10040150

Raschka, S. (2018). Model Evaluation, Model Selection, And Algorithm Selection In Machine Learning. *Arxiv Preprint Arxiv:1811.12808*. Https://Doi.Org/10.48550/Arxiv.1811.12808

Wafda, A. (2024). Integrasi Machine Learning Dalam Ritel: Tinjauan Komprehensif Tentang Prediksi Harga, Analisis Data Pelanggan, Dan Pemanfaatan Media Sosial. *Journal Artificial: Informatika Dan Sistem Informasi*, 2(2), 90–106. Https://Doi.Org/10.54065/Artificial.543

Wafda, A. (2025). Transfer Learning Advancements: A Comprehensive Literature Review On Text Analysis, Image Processing, And Health Data Analytics. *Journal Artificial: Informatika Dan Sistem Informasi*, 3(1), 1–10. Https://Doi.Org/10.54065/Artificial.544

Anggraini, D. A., Ikhsan, M., & Suhardi, S. (2024). Implementation Of The Naïve Bayes Algorithm In The Sms Spam Filtering System. *Journal Of Computer Networks, Architecture And High Performance Computing*, 6(2), 838-849. Https://Doi.Org/10.47709/Cnahpc.V6i2.3875

Krishnaveni, N., & Radha, V. (2021). Comparison Of Naive Bayes And Svm Classifiers For Detection Of Spam Sms Using Natural Language Processing. *Ictact Journal On Soft Computing*, *11*(2).

Setiyono, A., & Pardede, H. F. (2019). Klasifikasi Sms Spam Menggunakan Support Vector Machine. *Jurnal Pilar Nusa Mandiri*, *15*(2), 275-280. Https://Doi.Org/10.33480/Pilar.V15i2.693

Gupta, S. D., Saha, S., & Das, S. K. (2021). SMS Spam Detection Using Machine Learning. In *Journal Of Physics: Conference Series 1797(1)* 012017. IOP Publishing. Https://Doi.Org/10.1088/1742-6596/1797/1/012017

Singh, A. A. S., Tamilmani, V., Maniar, V., Kothamaram, R. R., Rajendran, D., & Namburi, V. D. (2021). Predictive Modeling For Classification Of SMS Spam Using NLP And ML Techniques. *International Journal Of Artificial Intelligence, Data Science, And Machine Learning*, 2(4), 60-69. Https://Doi.Org/10.63282/3050-9262.IJAIDSML-V2I4P107

Shahbazov, V. (2026). SMS Dataset For Multi-Class Classification Of Ham, Spam, And Smishing In Azerbaijani Language. *Problems Of Information Technology*, 32-39. Https://Doi.Org/10.25045/Jpit.V17.I1.04

Giri, S., Das, S., Das, S. B., & Banerjee, S. (2023). SMS Spam Classification–Simple Deep Learning Models With Higher Accuracy Using BUNOW And Glove Word Embedding. *Journal Of Applied Science And Engineering*, 26(10), 1501-1511. Https://Doi.Org/10.6180/Jase.202310_26(10).0015

Ugbotu, E. V., Aghaunor, T. C., Onoma, P. A., Max-Egba, A. T., Geteloma, V. O., Eboka, A. O., ... & Odiakaose, C. C. (2025). Transfer Learning Using A CNN Fused Random Forest For SMS Spam Detection With Semantic Normalization Of Text Corpus. *NIPES-Journal Of Science And Technology Research*, 7(2), 371-382. Https://Doi.Org/10.37933/Nipes/7.2.2025.29

Roy, P. K., Singh, J. P., & Banerjee, S. (2020). Deep Learning To Filter SMS Spam. *Future Generation Computer Systems*, *102*, 524-533. Https://Doi.Org/10.1016/J.Future.2019.09.001

Nagwani, N. K., & Sharaff, A. (2017). SMS Spam Filtering And Thread Identification Using Bi-Level Text Classification And Clustering Techniques. *Journal Of Information Science, 43(1),* 75-87. Https://Doi.Org/10.1177/0165551515616310