CoreID JOURNAL

Vol. 3 No. 3, November 2025 (114-123)
ISSN: 2987-6990
DOI: https://doi.org/10.60005/coreid.v3i3.145

Short Message Spam Classification using Decision Tree, Naive Bayes, and Logistic Regression

Azalia Fathimah Dinah[1], Citra Aulia[2], Dzilan Nazira Zahratunnisa[3], Rofik Efendi[4]
[1, 2, 3] Informatics Department, Faculty of Science and Technology, UIN Sunan Gunung Djati Bandung, Indonesia
[4]Sharia Economy Department, Faculty of Islamic Economics and Business, UIN Syekh Wasil Kediri, Indonesia

Article Info

Article history:

Received May 15, 2025
Revised October 27, 2025
Accepted November 26, 2025

Keywords:

Text Classification
Short Message Spam
Machine Learning
Decision Tree
Naïve Bayes
Logistic Regression
TF-IDF

ABSTRACT

The increasing use of Short Message Service (SMS) in digital communication has been accompanied by a rise in spam messages, which threaten user convenience and information security. This study presents a comparative analysis of three classical machine learning algorithms—Decision Tree, Naïve Bayes, and Logistic Regression—for SMS spam classification. The research follows the CRISP-DM methodology, including data collection, understanding, preparation, modeling, and evaluation. The dataset used is the *SMS Spam Collection (A More Diverse Dataset)* from Kaggle, comprising 5,574 SMS messages labeled as spam or ham. Text preprocessing is performed through cleaning operations and feature extraction using the Term Frequency–Inverse Document Frequency (TF-IDF) method. The models are evaluated using accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) metrics. Experimental results indicate that Logistic Regression achieves the most balanced performance, with an accuracy of 97.13%, precision of 99.23%, recall of 80.75%, F1-score of 89.04%, and an AUC of 98.72%. Naïve Bayes demonstrates high efficiency and perfect precision but lower recall, while Decision Tree offers interpretability with comparatively lower classification performance. The results suggest that Logistic Regression is the most suitable model for lightweight and reliable SMS spam detection systems, balancing accuracy and misclassification risk. This study provides practical insights for implementing efficient spam filtering solutions and serves as a reference for future research in text classification and natural language processing, particularly for short-message communication.

Corresponding Author:

Citra Aulia
Informatics Department, UIN Sunan Gunung Djati Bandung, Indonesia
Jl. AH. Nasution No. 105 Bandung
Email: citraaulia1812@gmail.com

1. INTRODUCTION

The rapid development of information and communication technology has significantly transformed the way people interact and exchange information. One communication medium that remains widely used is the *Short Message Service* (SMS). Despite the rapid growth of instant messaging applications such as WhatsApp and Telegram, SMS continues to play an important role, particularly in banking services, digital transactions, and the delivery of official notifications from various institutions [1].

Along with the extensive use of SMS, a serious problem has emerged in the form of SMS spam, which refers to unsolicited messages sent in bulk for commercial purposes, fraudulent activities, or other malicious intents. The presence of SMS spam not only disrupts user convenience but also poses potential threats to information security and financial safety. Therefore, an automated system capable of accurately and efficiently detecting and filtering spam messages is required [2].

Machine learning approaches have been widely applied to address this problem, especially in the field of text classification [3]. By utilizing labeled historical data, machine learning algorithms are able to learn

patterns and characteristics of spam messages, enabling the automatic classification of new messages into spam or non-spam (*ham*) categories [4].

Several classical machine learning algorithms have commonly been employed for SMS spam classification, including Decision Tree, Naive Bayes, and Logistic Regression [5], [6]. Decision Tree provides models that are easy to understand and interpret [7], Naive Bayes is known for its simplicity and effectiveness in text processing [8][9], while Logistic Regression demonstrates strong capability in modeling linear relationships between features and classification probabilities [3]. Nevertheless, each algorithm has its own strengths and limitations depending on the characteristics of the dataset and the preprocessing techniques applied.

Previous studies have demonstrated the effectiveness of these algorithms in SMS spam detection. Johari *et al.* reported that Decision Tree and Naive Bayes achieved high accuracy across various SMS spam datasets [10]. Yasmin and Aliza found that Naive Bayes was more efficient for short text messages, whereas Support Vector Machine (SVM) produced the highest accuracy [11]. Furthermore, studies by Theodorus *et al.* and publications presented at WCSE 2023 showed that classical models such as Naive Bayes, Logistic Regression, and Decision Tree remain competitive, particularly when trained on clean and representative datasets [12][13]. On the other hand, deep learning–based approaches such as IndoBERT have demonstrated superior accuracy but are less suitable for lightweight and resource-constrained real-world implementations [14].

Despite these findings, there is still a lack of studies that specifically compare the three classical algorithms—Decision Tree, Naive Bayes, and Logistic Regression—under consistent experimental conditions using more diverse and realistic datasets. Therefore, this study aims to conduct a comparative performance analysis of these three widely used machine learning algorithms for SMS spam classification [15]. The dataset used in this research was obtained from Kaggle, namely the *SMS Spam Collection (A More Diverse Dataset)*, which consists of text-based SMS messages labeled as spam or *ham*.

The results of this study are expected to identify the algorithm that delivers the best performance based on evaluation metrics such as accuracy, precision, recall, and F1-score. Moreover, this research is expected to contribute to the development of efficient and lightweight machine learning–based spam detection systems and to serve as a reference for future studies in the field of *Natural Language Processing* (NLP), particularly for short text messages and contexts relevant to the Indonesian language in digital communication services.

2. METHOD

The stages carried out in this study follow the CRISP-DM methodology, which begins with data collection, data understanding, data preparation, data modeling, and model evaluation [16], [17]. The explanation of each of these stages is presented in the following subsections.

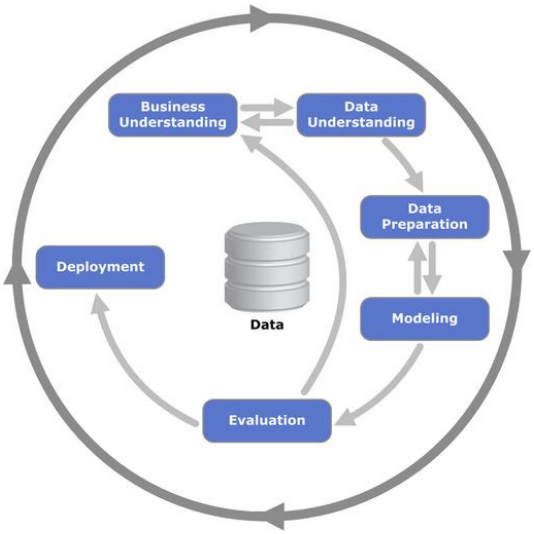

Figure 1. CRISP-DM Methodology

2.1 Collecting Data

The data were collected from the Kaggle website under the name SMS Spam Collection – A More Diverse Dataset. This dataset is a collection of short message service (SMS) messages consisting of 5,574 data instances with two labels categorized as spam or ham.

The dataset was designed to support text-based classification research and to represent real-world communication contexts. Unlike earlier and more limited versions of SMS spam datasets, this version is more diverse in terms of message formats and language styles.

The attributes of the data used in this study are described as follows.

Table 1. Data Attributes

Data Attributes	Description
sms	The raw SMS text message content, including informal language, abbreviations, punctuation, and free sentence structures commonly used in short message communication.
label	The category of the SMS message, classified as spam or ham.

2.2 Data Understanding

At this stage, several preliminary examinations were conducted to obtain a comprehensive understanding of the structure, quality, and key characteristics of the data to be used in the modeling process. This step is essential to ensure that the data meet the suitability criteria for machine learning–based classification and to anticipate potential issues that may affect model performance.

2.2.1 Data Structure Examination and Label Distribution

The dataset used consists of 5,574 records and two main columns, namely sms and label. The sms attribute contains raw short message service (SMS) text, which serves as the primary focus of the analysis, while the label attribute indicates the message category, where spam refers to unsolicited or promotional messages and ham refers to legitimate or normal messages. Initial exploratory analysis shows that the label distribution in the dataset is imbalanced, with 4,827 ham messages (approximately 86.6%) and 747 spam messages (approximately 13.4%).

2.2.2 Identification of Missing Values and Data Duplication

Data completeness validation was subsequently performed. The results indicate that there are no missing values in either column (sms or label), and therefore no imputation process was required. However, several duplicate messages were identified based on the content of the sms column. These duplicates could potentially amplify the weight of certain messages during training, which may lead to overfitting. Consequently, duplicate records were removed to maintain data diversity and improve model generalization.

2.2.3 Outlier Identification in Message Length Features

Although the dataset consists of textual data, a statistical analysis was conducted on message length in terms of character count and word count. The purpose of this analysis was to detect outliers or extreme values that could influence the weighting process in frequency-based models. The analysis revealed that some messages contain more than 500 characters, which is statistically far above the average message length. However, manual inspection confirmed that such messages are characteristic of long spam messages, such as repetitive promotional content or fraudulent formats. Therefore, these messages were retained in the dataset.

2.2.4 Correlation Analysis and Text Content Characteristics

To prepare the data for feature extraction using the TF-IDF approach, an initial analysis of the text content was performed, which included:

a) Dominant word frequency in each class;
b) Average message length;
c) The occurrence of specific keywords frequently found in spam messages, such as *"win"*, *"free"*, *"click"*, and *"claim"*.

The analysis results indicate that:

d) Spam messages tend to be longer, use promotional and manipulative vocabulary, and exhibit persuasive language patterns.
e) Ham messages have simpler and shorter structures and use more natural, everyday language.

These findings support the assumption that TF-IDF–based numerical representations are capable of capturing meaningful differences in word distribution between the spam and ham classes.

2.3 Data Preparation

This stage aims to transform raw data into a form that can be effectively used in the training process of a classification model. Given that the dataset consists of free-form and unstructured text, a series of preprocessing steps is required to ensure that the data representation meets the requirements of text-based machine learning models.

The data preparation stages carried out in this study are as follows:

1. Text Cleaning
   The first step is to clean text messages from irrelevant elements or those that may cause noise during the feature extraction process. This step results in cleaner and more consistent text. The processes include:
   a. Case folding: All characters in the messages are converted to lowercase to standardize word forms, so that "Free," "FREE," and "free" are treated as the same.
   b. URL removal: Link addresses such as http://, https://, and www. are removed because they are usually unique and do not provide consistent semantic information.
   c. Removal of numbers and punctuation: All numeric characters and punctuation marks (such as periods, commas, exclamation marks, and other symbols) are removed to simplify word representation.
   d. Removal of extra spaces: Multiple spaces and unnecessary whitespace characters are removed to make sentence structures more orderly.
2. Feature Extraction using TF-IDF
   After the text is cleaned, the next step is to convert it into a numerical representation using the TF-IDF (Term Frequency–Inverse Document Frequency) method. This method assigns weights to words based on:
   a. the frequency of a word's occurrence in a message (term frequency), and
   b. how rare the word is across all messages (inverse document frequency). TF-IDF produces a high-dimensional but sparse feature matrix of size n × m, where n is the number of SMS messages and m is the number of unique words. This representation allows the model to recognize important words such as "claim," "free," or "win," which appear more frequently in spam messages.
3. Data Labels
   In this dataset, the message category labels are already available in numerical form:
   a. 1 for the spam category, and
   b. 0 for the ham category.
   Therefore, no additional label encoding is performed, and these values are directly used as targets in the classification model training process.
4. Data Splitting (Train-Test Split)
   The dataset is then divided into two subsets: 80% is used as training data, and 20% is used as testing data. The split is performed using a stratified sampling technique to ensure that the proportions of spam and ham classes remain balanced in each subset.

2.4 Data Modelling

This stage aims to build and train a text-based classification model capable of distinguishing between spam and ham messages based on the numerical representation of SMS message content. Three supervised learning algorithms are employed in this study, namely Naive Bayes, Logistic Regression, and Decision Tree, each of which adopts different approaches and assumptions in processing textual data.

1. Pre-Modelling Data Preparation
   At this stage, the text data from the sms column are used as predictive features, while the label column serves as the target variable. The text messages have undergone basic cleaning processes, including case folding, removal of numbers, punctuation, and excessive whitespace, resulting in more uniform text that is ready for numerical representation.
   Numerical representation is performed using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. This method is selected because it assigns different weights to each word based on its local frequency within a message and its global frequency across the entire message corpus. Through this approach, words that are highly characteristic of a particular class—such as "claim," "win," "free," or "click" in spam messages—receive higher weights compared to common words that appear across all classes.
   To prevent data leakage, the fit_transform() process is applied only to the training data, while the test data are transformed using transform() with parameters learned exclusively from the training data. This step is crucial to maintain the integrity of model evaluation on previously unseen data.
   The dataset is split into 80% training data and 20% testing data using the train_test_split() function from the scikit-learn library.
2. Modelling with Naive Bayes, Decision Tree, and Logistic Regression
   a. Naive Bayes
      Naive Bayes is a probabilistic classification algorithm that is widely used in text classification tasks. This model assumes conditional independence among features (in this case, words) and operates based on Bayes' theorem to compute the probability of a class given the input features.
      In this study, the Multinomial Naive Bayes variant is employed, implemented using the MultinomialNB() function from the sklearn.naive_bayes library.

   b. Decision Tree
      Decision Tree is a non-linear, rule-based classification model. It operates by recursively partitioning the data based on the most informative features until homogeneous conditions are achieved at the leaf nodes.
      In this study, the DecisionTreeClassifier() from the sklearn.tree library is used with default parameters as a baseline model.

   c. Logistic Regression
      Logistic Regression is a linear classification model used to predict the probability of a target class. Despite its linear nature, this model is well suited for sparse data such as TF-IDF representations, as it can efficiently handle a large number of features.
      The model is implemented using LogisticRegression() from the sklearn.linear_model library. The max_iter parameter is increased to ensure convergence during the optimization process, considering the high dimensionality of text-based features.
      All three models are built and trained using the same training dataset. The prediction outputs on the test dataset are subsequently evaluated to compare the performance of the three classification models.

2.5 Evaluation

Model performance evaluation is conducted to measure the effectiveness of classification on the test data. The evaluation does not rely solely on accuracy but also employs more representative evaluation metrics. The primary metrics used include:

1. Accuracy: the percentage of correct predictions over the entire test dataset.
2. Precision: the proportion of messages predicted as spam that are actually spam.
3. Recall: the proportion of actual spam messages that are successfully identified by the model.
4. F1-score: the harmonic mean of precision and recall.

5. AUC-ROC: the area under the Receiver Operating Characteristic (ROC) curve, which measures the overall trade-off between the true positive rate (TPR) and the false positive rate (FPR).

The evaluation is performed using the classification_report() and roc_auc_score() functions from the sklearn.metrics library. Visualization of the confusion matrix is employed to explicitly present the number of correct and incorrect classifications. This visualization helps identify whether the model tends to produce more false positives or false negatives, which is particularly important in the context of spam detection.

3. RESULTS AND DISCUSSION

This study compares the performance of three machine learning algorithms—Decision Tree, Naïve Bayes, and Logistic Regression—in classifying SMS messages as spam or non-spam (ham). The evaluation is conducted using accuracy, precision, recall, F1-score, and AUC metrics, based on the "SMS Spam Collection (A More Diverse Dataset)" obtained from Kaggle. All models are trained using a uniform preprocessing approach and TF-IDF–based feature extraction techniques.

3.1. Model Comparation

Table 2. Comparation Result

Model	Accuracy	Precision	Recall	F1-score	AUC
Decision Tree	95,46%	85.59%	77.10%	81.12%	87.61%
Naive Bayes	96.33%	100%	72%	84%	97%
Logistic Regression	**97.13%**	**99.23%**	**80.75%**	**89.04%**	**98.72%**

Based on the comparison table of the three models, the Decision Tree model achieved an accuracy of 95.46%, precision of 85.59%, recall of 77.10%, and an F1-score of 81.12%. In addition, the AUC value reached 0.8761. The training time of this model was relatively fast, at approximately 0.8846 seconds. Although the accuracy is fairly high, the lower recall indicates that the model still struggles to identify some spam messages, leading to potential cases where spam is not properly filtered. This behavior suggests a tendency toward overfitting on the training data, particularly in the absence of pruning or further parameter optimization.

The Naïve Bayes model employed in this study uses the default implementation from the scikit-learn library, namely MultinomialNB. This model demonstrates good performance in text message classification, with a notably short training time of 0.0207 seconds. Evaluation on the test dataset resulted in an accuracy of 96.33%, precision of 100%, recall of 72%, and an F1-score of 84%. An AUC score of 97% further confirms the model's overall capability to distinguish between spam and non-spam messages.

While the high precision indicates that the model is highly accurate in detecting spam without producing many false positives, the lower recall shows that a portion of spam messages remains undetected. In other words, the model tends to be conservative when assigning the spam label, which negatively affects the F1-score. These results are consistent with the characteristics of the Naïve Bayes model, which is highly efficient but sensitive to data representation and feature distribution.

Logistic Regression exhibits the most balanced performance among the three models, achieving an accuracy of 97.13%, precision of 99.23%, recall of 80.75%, and an F1-score of 89.04%. The AUC value of 98.72% indicates an excellent ability to discriminate between spam and ham messages. Although some spam messages are still not detected, this model effectively minimizes false positives, making it well suited for spam filtering systems that prioritize classification accuracy.

Overall, the evaluation results indicate that Naïve Bayes excels in terms of efficiency and precision but tends to miss some spam messages due to its lower recall. In contrast, Logistic Regression demonstrates the best balance across all evaluation metrics, making it a strong candidate for implementation in short-text–based spam detection systems. Decision Tree, while not as strong as the other two models, remains relevant in scenarios that prioritize interpretability and fast training time.

Based on these findings, it can be concluded that Logistic Regression and Naïve Bayes are the most suitable models for SMS spam classification, particularly for lightweight and efficient systems that do not require substantial computational resources. The final choice of model can be adapted to the specific requirements of the system, whether the priority lies in overall accuracy, processing speed, or result interpretability.

3.2. Confusion Matrix Visualization

The Decision Tree model demonstrates fairly good performance in distinguishing between spam and ham messages. From the total test data, the model correctly classified 887 ham messages, while 17 ham messages were incorrectly identified as spam. For the spam category, 101 messages were correctly classified, whereas 30 spam messages were misclassified as ham. This indicates that the model has a strong ability to recognize legitimate (ham) messages but still tends to miss some spam messages, as reflected by a lower recall value compared to precision. In other words, the model is relatively more conservative in detecting spam, which poses a risk of allowing some harmful messages to pass through the filter. Figure 3 illlustrates the results of confusion matrix using Decision Tree.

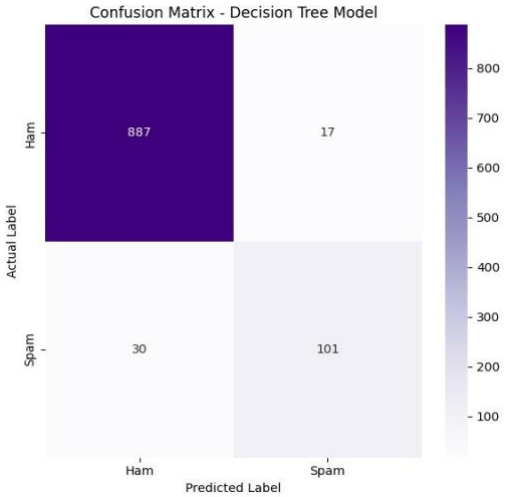

Figure 2. Confusion Matrix for Decision Tree

The distribution of these errors shows that the model remains vulnerable to false negatives, namely spam messages that go undetected. This suggests the need for parameter tuning or the application of pruning techniques to prevent overfitting to the training data.

The Naïve Bayes model successfully recognized all ham messages accurately (898 correct, 0 incorrect). However, only 99 out of 137 spam messages were correctly identified, while 38 spam messages were misclassified as ham. This model is highly precise in avoiding false positives (i.e., it does not incorrectly label legitimate messages as spam), but it is less sensitive to actual spam messages, resulting in a lower recall. This reflects the characteristics of Naïve Bayes as an efficient and fast model, yet one that tends to be conservative in spam detection. The model is suitable for systems that prioritize accuracy for important messages; however, caution is required since some spam messages may still go undetected. The result of confusion matrix shows on the figure 2.

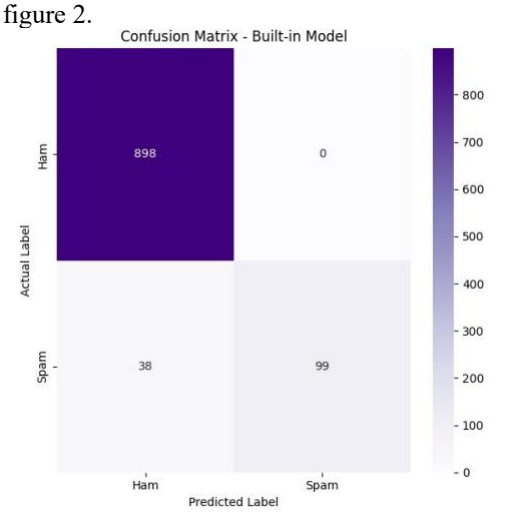

Figure 3. Confusion Matrix for Naive Bayes

The Logistic Regression model demonstrates the most balanced performance. From the entire test dataset, the model correctly identified 953 ham messages, with only 1 ham message misclassified as spam. On the spam side, 130 messages were correctly detected, while 31 spam messages were missed. This model

exhibits a very low false positive rate and a relatively high recall, making it ideal for spam detection systems that require accuracy and balanced classification. These results indicate that Logistic Regression has strong detection capabilities for both classes, spam and ham, with a very low error rate. The confusion matrix of logistic regression shows on figure 3.

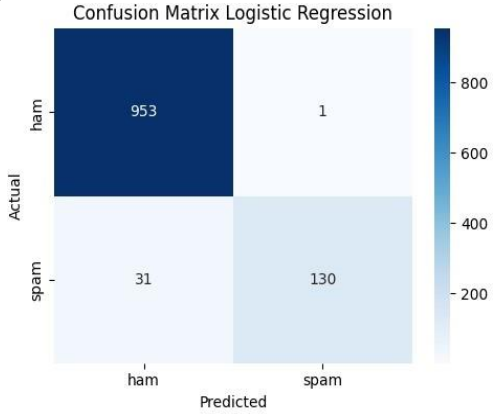

Figure 4. Confusion Matrix for Logistic Regression

3.3. Interpretation of Results and Model Implications

The evaluation results of the three classification models reveal fundamental differences in their decision-making approaches, which directly affect spam message classification patterns. In the context of short, ambiguous, and highly variable text data such as SMS messages, model effectiveness is strongly influenced by how features are extracted and how the model handles informational irregularities.

The varying performance across metrics such as recall and precision indicates that no single algorithm is optimal in all aspects. In spam detection, where system effectiveness largely depends on the consistent identification of harmful messages, the trade-off between false positives and false negatives becomes critical. Models that are overly cautious in labeling messages as spam may fail to detect real threats, while overly aggressive models may disrupt users by misclassifying legitimate messages as spam. Therefore, model selection should be aligned with the risk context and tolerance level of the intended system.

Training time efficiency is also an important consideration. In system environments that require rapid processing, such as real-time message filtering for SMS banking services or emergency notifications, models with low inference time and low computational complexity are highly valuable. This demonstrates that algorithm effectiveness should not be measured solely by accuracy, but also by how efficiently it can be integrated into existing system infrastructures.

Furthermore, these findings reinforce the view that data representation quality plays a decisive role in text classification success. While TF-IDF-based approaches help balance the influence of common and specific terms, they remain limited in capturing deeper semantic meaning. To address this limitation, embedding-based or contextual modeling approaches may serve as promising directions for future development to improve model sensitivity to increasingly complex spam variations.

From an implementation perspective, Logistic Regression shows strong potential as a stable and well-balanced model for production applications, while Naïve Bayes excels in speed and resource efficiency. Decision Tree models, although sensitive to data variation, remain a viable option in systems that require transparent decision logic.

The results of this study provide a strong foundation for selecting spam classification models for real-world applications. System developers who prioritize speed and efficiency may rely on Naïve Bayes, whereas systems that demand balanced classification and interpretability may consider Logistic Regression. Deployment in critical systems such as banking services or official notification platforms can be tailored based on tolerance levels for false positives and false negatives.

This study is limited to three classical models and a single feature representation approach. Future exploration of ensemble models, boosting techniques, and deep learning-based approaches such as BERT could provide broader insights into modern spam classification potential. Additionally, evaluating model performance under imbalanced data conditions or in multilingual message scenarios remains an important area for further research. Testing in real-time and streaming data scenarios would further strengthen the contribution of these findings toward the development of adaptive and robust spam detection systems.

4. CONCLUSION

This study compares three text classification algorithms—Decision Tree, Naïve Bayes, and Logistic Regression—in detecting spam messages in SMS using the SMS Spam Collection dataset (A More Diverse

Dataset). The evaluation results indicate that all three models demonstrate fairly good performance, although each exhibits distinct strengths.

Based on the experimental results, Logistic Regression shows the most balanced performance, achieving an accuracy of 97.13%, precision of 99.23%, recall of 80.75%, and an F1-score of 89.04%. The model also records an AUC of 98.72%, indicating its consistent ability to distinguish between spam and ham messages. Naïve Bayes, on the other hand, excels in training time efficiency and achieves very high precision (100%), although its recall is relatively low (72%), reflecting the model's tendency to be more conservative in spam detection. The Decision Tree model offers high interpretability but exhibits weaker recall (77.10%) and a lower AUC (87.61%) compared to the other two models.

Considering the evaluation results and the confusion matrix visualization, Logistic Regression can be concluded as the most suitable model for the SMS spam classification task, as it achieves a good balance between detecting spam messages and minimizing misclassification of legitimate (ham) messages. This model is well suited for deployment in production systems that require high accuracy, stability, and strong generalization capability.

Future research may focus on using larger and more diverse datasets, including multilingual and multi-platform data, to improve model generalization. Ensemble learning methods such as Random Forest, AdaBoost, and XGBoost could be explored to enhance accuracy and reduce overfitting. Additionally, deep learning approaches such as RNN and BERT may be investigated to capture richer semantic context. Evaluating models in real-time or streaming data scenarios is also recommended to assess performance in dynamic environments. The machine learning–based spam detection system has the potential to become a more adaptive, efficient, and robust solution for handling the continually evolving variety of message types.

REFERENCES

[1] S. K. D. Sharma, "A Comparative Study of Machine Learning Classifiers for Different Language Spam SMS Detection: Performance Evaluation and Analysis," *Advances in Artificial Intelligence Research*, vol. 4, no. 2, pp. 69–77, 2024, doi: 10.54569/aair.1549781.

[2] D. A. Oyeyemi and A. K. Ojo, "SMS Spam Detection and Classification to Combat Abuse in Telephone Networks Using Natural Language Processing," *Journal of Advances in Mathematics and Computer Science*, vol. 38, no. 10, pp. 144–156, Oct. 2023.

[3] S. Kaddoura, G. Chandrasekaran, D. E. Popescu, and J. H. Duraisamy, "A Systematic Literature Review on Spam Content Detection and Classification," *PeerJ Comput. Sci.*, vol. 8, p. e830, 2022.

[4] M. Ahmadi and others, "Leveraging Large Language Models for Cybersecurity: Enhancing SMS Spam Detection with Robust and Context-Aware Text Classification," *arXiv preprint*, vol. arXiv:2502.11014, 2025, [Online]. Available: https://arxiv.org/abs/2502.11014

[5] M. R. Al Saidat, S. Y. Yerima, and K. Shaalan, "Advancements of SMS Spam Detection: A Comprehensive Survey of NLP and ML Techniques," *Procedia Comput. Sci.*, vol. 244, pp. 248–259, Jan. 2024, doi: 10.1016/J.PROCS.2024.10.198.

[6] M. A. Abid, S. Ullah, M. A. Siddique, M. F. Mushtaq, W. Aljedaani, and F. Rustam, "Spam SMS filtering based on text features and supervised machine learning techniques," *Multimedia Tools and Applications 2022 81:28*, vol. 81, no. 28, pp. 39853–39871, May 2022, doi: 10.1007/S11042-022-12991-0.

[7] A. A. Dehghani, N. Movahedi, K. Ghorbani, and S. Eslamian, "Decision tree algorithms," *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*, pp. 171–187, Jan. 2023, doi: 10.1016/B978-0-12-821285-1.00004-X.

[8] Y. Ying, T. N. Mursitama, Shidarta, and Lohansen, "Effectiveness of the News Text Classification Test Using the Naïve Bayes' Classification Text Mining Method," *J. Phys. Conf. Ser.*, vol. 1764, no. 1, p. 012105, Feb. 2021, doi: 10.1088/1742-6596/1764/1/012105.

[9] W. B. Zulfikar, A. R. Atmadja, and S. F. Pratama, "Sentiment Analysis on Social Media Against Public Policy Using Multinomial Naive Bayes," *Scientific Journal of Informatics*, vol. 10, no. 1, pp. 25–34, Jan. 2023, doi: 10.15294/SJI.V10I1.39952.

[10] M. F. Johari, M. A. A. Ghani, S. Z. M. Hashim, and A. M. Kamaruddin, "Key Insights into Recommended SMS Spam Detection Datasets," *Sci. Rep.*, vol. 15, no. 1, p. 8162, Mar. 2025, [Online]. Available: https://www.nature.com/articles/s41598-025-92223-1

[11] H. Y. Aliza and H. Yasmin, "A Comparative Analysis of SMS Spam Detection Employing Machine Learning Methods," *Int. J. Comput. Appl.*, vol. 182, no. 20, pp. 15–21, Mar. 2022, [Online]. Available: https://www.researchgate.net/publication/359576155

[12] T. E. B. Theodorus, Y. Yanuar, and R. D. Wulandari, "Short Message Service (SMS) Spam Filtering Using Deep Learning in Bahasa Indonesia," in *Proceedings of the 4th International Conference on Information and Communications Technology (ICoICT)*, Yogyakarta, Indonesia, 2021, pp. 104–109.

[13] K. Ahluwalia, G. H L, R. R, and H. Lin, "Comparative Analysis of Various SMS Spam Detection Techniques Using Machine Learning Algorithms," in *Proceedings of the 13th International Workshop on Computer Science and Engineering (WCSE)*, 2023, pp. 87–92. [Online]. Available: https://www.wcse.org/WCSE_2023/021.pdf

[14] N. Latifah, R. Dwiyansaputra, and G. S. Nugraha, "Multiclass Text Classification of Indonesian Short Message Service (SMS) Spam using Deep Learning Method and Easy Data Augmentation," *MATRIK: Jurnal*

Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 23, no. 3, pp. 663–676, Jul. 2024, [Online]. Available: https://www.researchgate.net/publication/382758470

[15] Q. Li *et al.*, "A Survey on Text Classification: From Shallow to Deep Learning," *arXiv preprint*, vol. arXiv:2008.00364, 2021, [Online]. Available: https://arxiv.org/abs/2008.00364

[16] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," *Procedia Comput. Sci.*, vol. 181, pp. 526–534, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.199.

[17] M. Irfan, T. V. Riyadi, A. R. Atmadja, R. S. Fuadi, and A. Muin, "Application of Convolutional Neural Network Algorithm for Analyzing Sentiments on the Kampus Merdeka Policy," *Proceeding of 2024 the 10th International Conference on Wireless and Telematics, ICWT 2024*, 2024, doi: 10.1109/ICWT62080.2024.10674724.