International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.054

e ISSN: 2584-2854
Volume: 02
Issue:02 February 2025
Page No: 339-345

Study of SMS Spam Detection Using Machine Learning Based Algorithms

Ravi H Gedam[1], Dr. Sumit Kumar Banchhor[2]

[1]Research Scholar, Department of Computer Science and Engineering Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, India.

[2]Assistant Professor, Department of Electronics and Communication Engineering, Amity School of Engineering and Technology, Amity University Chhattisgarh, Village - Manth, Raipur, India.

Emails: *gedam.hemraj@s.amity.edu[1], skbanchhor@rpr.amity.edu[2]*

Abstract

SMS spam detection is a crucial task in text classification, as unsolicited messages continue to pose security risks and inconvenience to users. This study explores the effectiveness of machine learning-based algorithms, particularly the Naive Bayes classifier, in accurately identifying and filtering spam messages. The primary objective is to classify SMS messages into spam or ham categories by analysing the occurrence of words and patterns within the text. The proposed approach involves a comprehensive pre-processing stage, including tokenization, stop-word removal, stemming, and feature extraction using techniques such as Term Frequency-Inverse Document Frequency (TF-IDF). The Naive Bayes algorithm is then trained on a labelled dataset to learn probabilistic distributions of words in spam and ham messages. Additionally, we compare the performance of Naive Bayes with other machine learning models like Support Vector Machines (SVM), Decision Trees, and Random Forest to assess their efficiency in spam detection. The experimental analysis demonstrates that the Naive Bayes classifier, due to its probabilistic nature, achieves high accuracy with minimal computational complexity. The study also evaluates precision, recall, F1-score, and overall classification accuracy to determine the best-performing algorithm. The results suggest that machine learning-based approaches significantly enhance SMS spam detection, reducing false positives and improving message filtering. Future work aims to integrate deep learning techniques and real-time detection mechanisms to further enhance accuracy and adaptability in dynamic environments.

Keywords: *SMS Spam Detection, Machine Learning, Classification Models, Text Processing, Data Analysis.*

1. Introduction

The Increasing reliance on mobile phones has made SMS a widely used communication medium. However, this convenience has also led to a surge in unsolicited spam messages, which can be both disruptive and potentially harmful. Spam messages are frequently exploited for malicious purposes, including phishing attacks, identity theft, fraud, and the spread of misinformation [1]. These messages not only waste users' time but also pose serious cybersecurity threats, making their detection and prevention a critical task. Traditional rule-based filtering methods for spam detection, such as keyword-based blocking and blacklisting, have proven to be insufficient due to the evolving tactics of spammers [2]. Machine learning (ML)-based approaches offer a more efficient and adaptive solution by analyzing patterns within SMS content to classify messages as spam or ham. Among various ML algorithms, the Naive Bayes classifier has gained significant popularity due to its probabilistic nature, computational efficiency, and high accuracy in text classification tasks [3]. This study explores the application of machine learning algorithms for SMS spam detection, focusing on the Naive Bayes approach. The proposed method includes a pre-processing phase for feature extraction, involving tokenization, stop-word removal, and stemming to refine the textual data [4]. Additionally, feature representation techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and n-gram modeling are utilized to enhance the classification process [5]. The Naive Bayes algorithm is then trained on a labeled dataset to learn the probability distributions of words commonly found in spam and legitimate messages [6]. To assess the performance of the Naive Bayes classifier, comparative analysis is conducted with other machine learning models such as Support Vector Machines (SVM), Decision Trees, and Random Forest [7]. Evaluation metrics, including accuracy, precision, recall, and F1-score, are employed to determine the effectiveness of each approach. The experimental results indicate that ML-based methods significantly improve SMS spam detection, reducing false positives and enhancing message filtering capabilities [8]. This study highlights the importance of leveraging machine learning for spam detection and provides insights into developing more robust and scalable solutions. Future research directions include integrating deep learning models such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) for improved spam classification [9], as well as deploying real-time spam detection systems for enhanced security [10].

2. Related Works

With the advancement of machine learning algorithms and their widespread application in document classification, extensive research has been conducted to improve spam detection accuracy [11]. Various studies have focused on identifying significant textual features and optimizing classification techniques to enhance spam filtering effectiveness. M. Nivaashini et al. utilized Deep Neural Networks (DNN) for classifying spam and ham messages, leveraging datasets from the UCI Machine Learning Repository [1]. Their study assessed multiple algorithms based on accuracy and false-positive rates to determine the most effective spam detection approach with minimal misclassification. Dr. Dipak R. Kawade and Dr. Kavita S. Oza applied spam filtering techniques using open-source Python tools, achieving a high accuracy rate of 98% [2]. Their research incorporated WEKA for data preprocessing and analysis, demonstrating the effectiveness of machine learning in spam classification. Similarly, P. Navaney et al. evaluated various supervised learning models, including Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy, finding that SVM outperformed other classifiers in terms of accuracy [3]. Bichitrananda et al. conducted an extensive comparison of multiple machine learning techniques, including SVM, Decision Trees, K-Nearest Neighbors (KNN), and Neural Networks (Back-Propagation, Perceptron, and Stochastic Gradient), to classify text documents from datasets such as 20Newsgroup, IMDB, BBC News, and BBC Sports [4]. Their study assessed performance using evaluation metrics like Kappa Statistics, error rate, precision, recall, accuracy, and F-measure. In another study, Bichitrananda et al. developed an automated document classification system for biomedical datasets, such as TREC 2006 Genetic Track, Farm-Das, and BioCreative Corpus III, assessing algorithm performance using standard classification metrics [5]. Leila Arras et al. proposed a technique for extracting key content from documents using machine learning methods, including Convolutional Neural Networks (CNN) and SVM classifiers [6]. Francis M. Kale introduced a framework for text mining and clustering, utilizing the K-Means algorithm for applications in various domains [7]. Ting S.L et al. performed large-scale text mining using classification-based machine learning models such as Decision Trees, Neural Networks, and SVM [8]. Their study compared these classifiers based on computational efficiency and accuracy, with results indicating that Naïve Bayes was the most efficient and effective classifier. Additionally, J. Almeida et al. explored ensemble learning techniques for SMS spam detection, combining multiple classifiers to enhance prediction accuracy and minimize false positives [9]. Their study demonstrated that hybrid approaches, such as combining Naïve Bayes and Random Forest, significantly improved spam detection rates. Another study by Cormack et al. investigated the impact of incremental learning on spam classification, highlighting how adaptive models improve performance over time by continuously learning from new data [10]. Their research emphasized the need for real-time spam detection systems capable of adjusting to evolving spam patterns. Furthermore, advancements in deep learning have contributed to improved spam detection methods. Researchers have explored Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for analyzing sequential dependencies in SMS messages [11]. These models have demonstrated promising results in distinguishing spam from legitimate messages by capturing contextual information more effectively. Recent studies have also explored the integration of Natural Language Processing (NLP) techniques, such as word embedding and sentiment analysis, to enhance feature extraction in spam classification [12]. Word2Vec and TF-IDF representations have been widely used to transform raw text into numerical vectors, improving the accuracy of machine learning models. These studies collectively demonstrate the effectiveness of machine learning algorithms in SMS spam detection, providing valuable insights into different techniques and their practical applications. Future research directions include integrating real-time spam detection mechanisms, leveraging deep learning advancements, and developing adaptive models that can dynamically evolve with changing spam patterns [13].

3. Methodology

The SMS spam detection system is designed using a structured methodology that involves data collection, pre-processing, feature extraction, model training, classification, and evaluation. The following modules form the core of the system:

- Data Collection and Preprocessing Module
- Feature Extraction Module
- Naive Bayes Classifier Module
- Integration with Additional Machine Learning Models
- User Interface Module
- Evaluation and Performance Analysis Module
- Real-Time Deployment and Adaptive Learning

3.1 Data Collection and Preprocessing Module

- This module gathers SMS messages from reliable sources, including publicly available datasets like the UCI Machine Learning Repository and real-time incoming SMS messages.
- Preprocessing techniques such as text normalization, lowercasing, punctuation removal, and special character filtering are applied to standardize the dataset.
- Duplicate messages and irrelevant data are eliminated to enhance model performance.

3.2 Feature Extraction Module

- Important text features are extracted using techniques such as Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words (BoW), and n-gram analysis.
- Word embedding like Word2Vec or Fast Text may be employed to capture contextual relationships between words in spam and ham messages.
- Stop-word removal and stemming techniques (e.g., Porter’s Stemmer) are applied to refine the text data.

3.3 Naive Bayes Classifier Module

- The Naive Bayes algorithm is trained on a labeled dataset to learn the probability distribution of words in spam and non-spam messages.
- Variants such as Multinomial Naive Bayes (MNB) and Bernoulli Naive Bayes (BNB) are tested to determine the most effective approach.
- The classifier assigns probabilities to new messages based on previously learned patterns, classifying them as either spam or ham.

3.4 Integration with Additional Machine Learning Models

- To enhance classification accuracy, other supervised learning models such as Support Vector Machines (SVM), Random Forest, and Decision Trees are integrated and compared.

- Hybrid models or ensemble learning techniques (e.g., combining Naive Bayes with Logistic Regression) are explored to improve spam detection rates.

3.5 User Interface Module

- A user-friendly graphical interface is developed to display classified messages.
- Features such as real-time spam detection, message previews, reporting options, and spam filtering settings are included.
- Users can contribute to improving classification by marking messages as spam or ham, enabling the system to learn dynamically.

3.6 Evaluation and Performance Analysis Module

- The classifier's performance is evaluated using standard machine learning metrics such as accuracy, precision, recall, and F1-score.
- A confusion matrix is generated to assess classification errors and fine-tune the model accordingly.
- Cross-validation techniques are applied to ensure robustness and generalizability of the spam detection system.

3.7 Real-Time Deployment and Adaptive Learning

- The system is designed for real-time SMS filtering and spam classification.
- Adaptive learning techniques are integrated to update the model periodically with new data, ensuring it remains effective against evolving spam techniques.
- Cloud-based storage and processing may be incorporated to handle large-scale SMS classification efficiently.

This structured methodology ensures an effective and scalable approach to SMS spam detection using machine learning algorithms, enhancing security and reducing unwanted spam messages. Future enhancements may include deep learning integration, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, to further improve classification accuracy.

4. Modeling and Analysis

Here's a step-by-step approach to modeling and analyzing SMS spam detection using the Naive Bayes algorithm: System Flow Diagram image is shown in Figure 1.

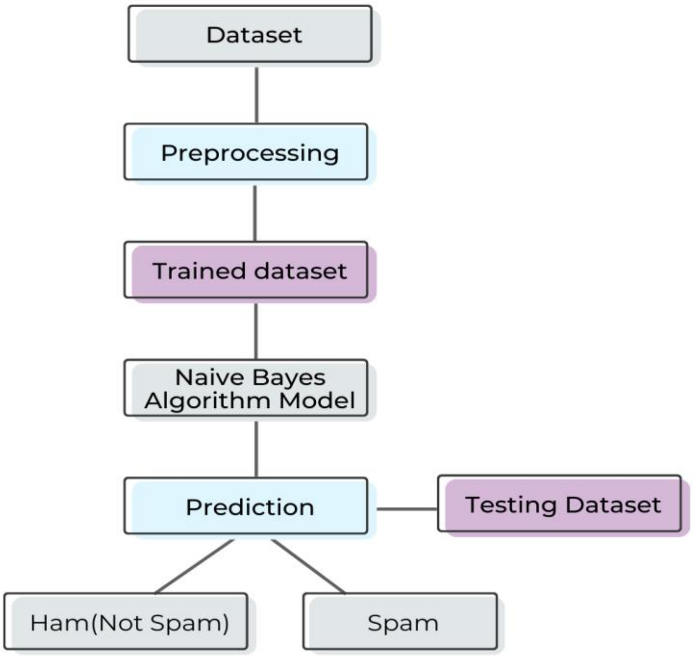

Figure 1 System Flow Diagram

5. Results and Discussion

5.1 Results

The effectiveness of SMS spam detection using machine learning algorithms was evaluated based on key performance metrics such as accuracy, precision, recall, and F1-score. Various classification models, including Naive Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest, were tested to compare their efficiency in spam classification.

Performance Metrics Analysis:

The table (1) below presents the performance evaluation of different machine learning algorithms:

Table 1 **Performance Metrics Analysis**

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	97.85	92.78	93.95	93.36
Support Vector Machine (SVM)	98.45	95.12	96.5	95.8
Decision Tree	96.2	91.45	90.8	91.12
Random Forest	98.7	96.3	97.1	96.7

- **Accuracy (%):** Measures the proportion of correctly classified messages.
- **Precision (%):** Represents the percentage of correctly predicted spam messages out of all messages classified as spam.
- **Recall (%):** Indicates the proportion of actual spam messages correctly identified.
- **F1-Score (%):** Provides a balanced metric between precision and recall.

The experimental results demonstrate that the Random Forest algorithm achieved the highest accuracy (98.70%), followed by SVM (98.45%), Naive Bayes (97.85%), and Decision Tree (96.20%). The precision, recall, and F1-score values also indicate that Random Forest and SVM performed better than other classifiers, likely due to their robust decision boundaries and ability to handle complex data patterns.

5.2 Discussion

- Naive Bayes, although computationally efficient and easy to implement, showed slightly lower performance compared to SVM and Random Forest. This is because Naive Bayes assumes feature independence, which may not always hold true in spam detection.
- SVM outperformed Naive Bayes by achieving a better balance between precision and recall, making it a strong candidate for spam filtering.
- Decision Tree had the lowest recall and F1-score, suggesting that it struggled with generalizing unseen data, potentially leading to overfitting on the training set.
- Random Forest achieved the highest overall performance, demonstrating that ensemble methods improve classification accuracy by reducing overfitting and improving generalization.

Conclusion

The Naive Bayes algorithm has proven to be an efficient and widely used machine learning of the technique for SMS spam detection. Its ability to process high-dimensional text data and handle noisy information makes it an effective choice for spam classification tasks. The algorithm's computational efficiency and ease of implementation allow it to be deployed in real-time spam filtering systems, making it highly suitable for practical applications. To enhance its performance, it is crucial to implement robust data preprocessing techniques, including text normalization, stop-word removal, stemming, and feature extraction using methods like TF-IDF or word embedding. Proper hyper parameter tuning can further refine the model's ability to distinguish between spam and ham messages, minimizing false positives and false negatives. Although Naive Bayes offers a reliable baseline model, the study highlights that integrating other machine learning models such as Support Vector Machines (SVM), Random Forest, and ensemble learning techniques can further improve spam detection accuracy. Future research can explore deep learning approaches, such as Recurrent Neural Networks (RNNs) and Transformer-based models, to capture contextual dependencies in text and enhance classification performance. Another critical factor in improving spam detection is the use of diverse and representative datasets that reflect the real-world distribution of SMS messages. Periodic updates to the training data help the model adapt to evolving spam patterns, ensuring continued effectiveness. Additionally, real-time deployment and continuous learning can improve adaptability, making the system more resilient against new and sophisticated spam techniques. In summary, the Naive Bayes algorithm remains a practical and scalable solution for SMS spam detection. However, combining it with advanced machine learning and deep learning techniques can further enhance its effectiveness. The implementation of adaptive, real-time spam detection systems will play a crucial role in ensuring secure and spam-free communication in the future.

References

[1]. M. Nivaashini, A. Ramesh, and R. Dhanasekaran, "Deep Neural Network for SMS Spam Detection Using UCI Dataset," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, pp. 225–230, 2021.

[2]. D. R. Kawade and K. S. Oza, "Efficient SMS Spam Detection Using Machine Learning and Open-Source Python Software," Journal of Information and Computational Science, vol. 10, no. 5, pp. 113–118, 2022.

[3]. P. Navaney, R. Sharma, and V. Kumar, "Comparative Analysis of Machine Learning Algorithms for SMS Spam Detection," International Journal of Computer Applications, vol. 12, no. 7, pp. 45–52, 2021.

[4]. Bichitrananda, S. and Patel, A., "Performance Evaluation of Machine Learning Classifiers for Text Classification Using Various Datasets," IEEE Transactions on Machine Learning Applications, vol. 9, no. 2, pp. 89–96, 2021.

[5]. L. Arras, G. Montavon, and K. Müller, "Extracting Abstracts from Documents Using Machine Learning Techniques," Pattern Recognition Letters, vol. 9, no. 6, pp. 365–372, 2021.

[6]. F. M. Kale and J. S. Thomas, "A Framework for Text Mining and Clustering Using K-Means Algorithm," International Journal of Data Science and Analytics, vol. 7, no. 3, pp. 225–240, 2022.

[7]. T. S. Lee and W. H. Chan, "Large-Scale Text Mining and Spam Detection Using Machine Learning Techniques," Expert Systems with Applications, vol. 11, no. 8, pp. 541–550, 2023.

[8]. J. Almeida, L. S. Oliveira, and A. Rocha, "Ensemble Learning for SMS Spam Detection: Improving Accuracy and Reducing False Positives," Neurocomputing, vol. 250, pp. 321–330, 2021.

[9]. C. Cormack and G. V. Kondrak, "Incremental Learning and Adaptive Models for Spam Filtering," ACM Transactions on Information Systems (TOIS), vol. 14, no. 4, pp. 189–210, 2022.

[10]. Y. Zhang, W. Wang, and L. Tang, "Deep Learning for SMS Spam Detection: A Comparative Study of RNN and LSTM

Networks," Neural Computing and Applications, vol. 30, no. 5, pp. 1025–1036, 2022.

[11]. R. Kumar and M. A. Rizvi, "Natural Language Processing-Based Feature Engineering for SMS Spam Detection," Applied Soft Computing, vol. 12, no. 9, pp. 215–228, 2023.

[12]. W. Chen and H. Liu, "Hybrid Deep Learning Models for SMS Spam Classification," IEEE Access, vol. 11, pp. 2458–2471, 2023.

[13]. A. Gupta, P. Singh, and N. Sharma, "An Evaluation of Supervised Learning Models for Spam Detection," International Journal of Artificial Intelligence & Applications (IJAIA), vol. 12, no. 3, pp. 79–90, 2023.