SMS Guard: Enhancing Spam Detection Using Multinomial Naive Bayes for Secure Communication

Manjula Devi C
Department of Information Technology
Velammal College of Engineering and Technology
Madurai, India
cmd@vcet.ac.in

Gobinath A
Department of Information Technology
Velammal College of Engineering and Technology
Madurai, India
agn@vcet.ac.in

Pooranalakshmi C
Department of Information Technology
Velammal College of Engineering and Technology
Madurai, India
pooranalakshmi2004@gmail.com

Samhita M
Department of Information Technology
Velammal College of Engineering and Technology
Madurai, India
samhita162004@gmail.com

Subashini G
Department of Information Technology
Velammal College of Engineering and Technology
Madurai, India
subashiniganeshan22@gmail.com

***Abstract*— This paper presents a robust spam detection system, SMS Guard, that leverages the Multinomial Naive Bayes algorithm to classify SMS messages as either spam or ham. With the increasing volume of SMS traffic, spam messages have become a significant threat to users, leading to privacy breaches and online fraud. The proposed model is trained on a labeled dataset containing both legitimate (ham) and spam messages. Through efficient preprocessing techniques, such as tokenization and normalization, the model achieves high accuracy and precision in detecting spam messages. By evaluating the system's performance using metrics like F1 score, this paper demonstrates the effectiveness of the model in providing a reliable solution for real-time spam detection, enhancing user security in digital communication.**

***Keywords*— SMS Spam Detection, Multinomial Naive Bayes, Machine Learning, Text Classification, Ham and Spam Messages**

I. INTRODUCTION

SMS (Short Message Service) remains a critical mode of communication in the increasingly interconnected world of today, with billions of messages exchanged daily on a global scale. In spite of the proliferation of alternative messaging platforms, SMS continues to be a popular choice for both personal and professional communication due to its accessibility and simple interface. However, the volume of SMS traffic has increased in tandem with the rise in the number of unsolicited and malicious messages, which are commonly referred to as "spam." These spam messages are not merely an inconvenience; they frequently present substantial security risks, such as links to malicious websites, fraudulent schemes, and phishing attempts. It is becoming increasingly challenging for users to distinguish between genuine and fraudulent communications due to the increasing sophistication of spam messages, which can occasionally appear to originate from legitimate sources [1].

In the past, spam detection systems have utilized rule-based methods, which employ predefined filters and patterns to identify spam messages. Despite their effectiveness in the past, these methods have become less dependable as spammers continue to develop their strategies to circumvent these filters. For example, spam messages may contain misleading URLs, altered orthography, or the use of symbols, which complicate the detection of the messages by conventional systems. In response to these challenges, machine learning (ML) techniques have emerged as a promising alternative. ML-based systems have the advantage of learning from large datasets, adapting to new spam patterns, and providing more accurate and scalable solutions [2].

This paper presents SMS Guard, a sophisticated spam detection system that leverages the Multinomial Naive Bayes (MNB) algorithm for classifying SMS messages as either spam or ham (legitimate messages). The Naive Bayes algorithm is particularly well-suited for text classification tasks due to its effective management of large volumes of data, speed, and simplicity.

The motivation behind SMS Guard is to provide users with an enhanced measure of security in their SMS communications. With increasing instances of SMS-based cybercrimes, such as phishing and identity theft, there is a compelling need for effective spam detection systems. These offenses often lead to significant financial losses and compromise sensitive personal information. Therefore, the implementation of real-time spam filtering solutions is essential to safeguard consumers from potential threats. Furthermore, this paper evaluates the performance of the proposed system using critical metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the efficacy of the model in accurately identifying spam messages with minimal false positives, thus providing a reliable solution for real-time spam detection. Additionally, the simplicity of the Naive Bayes approach assures that the system is computationally efficient, making it suitable for deployment on mobile devices with limited processing power [3].

In this research seeks to contribute to the growing field of SMS spam detection by offering a robust, machine-learning-based solution that not only detects spam with high accuracy but also adapts to the evolving nature of threats in the digital communication space. Through SMS Guard, we provide a scalable, lightweight system that enhances user protection in an increasingly vulnerable communication medium.

II. Related Work

The increasing threat of spam across different communication platforms has led to significant research efforts aimed at developing effective spam detection models. Various machine learning techniques have been applied to address the growing challenge of filtering unsolicited messages in real-time, improving user security, and preventing fraud. This section explores key research contributions in the field of SMS and communication spam detection, highlighting different approaches and methodologies.

Reissbrodt, Suleiman Y. Yerima, and Abul Bashar, in their paper *"Semi-supervised Novelty Detection with One-Class SVM for SMS Spam Detection,"* propose a model that uses labeled and unlabeled SMS datasets to identify spam. While most messages are unlabeled, a small set of legitimate (ham) messages is used to train the model. Key spam features include text content, message length, hyperlinks, and keywords. The Support Vector Machine (SVM) constructs a decision boundary to capture legitimate data while excluding outliers, making it suitable for mobile applications and messaging systems to enhance spam protection [4].

In *"Detection of Spam Messages in E-Messaging Platform Using Machine Learning,"* S. Vinothkumar et al. address spam detection on e-messaging platforms, focusing on user safety. They employ Naive Bayes and SVM algorithms trained on spam and ham messages to identify distinguishing patterns. The developed model analyzes incoming messages in real time, applicable for SMS filtering, email spam detection, and social media monitoring [5].

Anuj Gupta's paper, *"Detection of Spam and Fraudulent Calls Using Natural Language Processing (NLP),"* emphasizes identifying fraudulent calls for user protection. The model is trained on call transcripts with both legitimate and fraudulent examples, extracting features like text content and sentiment analysis. NLP algorithms, including logistic regression and deep learning, classify real-time call transcripts to alert users about potential scams [6].

Mustafa Al-farttoosi and Hasan Abdulkader discuss in *"How Botnets Become Secure and Menacing Threats to Mobile Devices"* the use of machine learning in detecting botnets. Their model is trained on logs of legitimate and compromised device behavior, using traditional models like decision trees and SVMs for initial classification and advanced models like recurrent neural networks (RNNs) for capturing complex patterns in real time, minimizing false positives [7].

In *"Detection of Multilingual Spam SMS Using Naive Bayes Classifier,"* Aparna K and Sayan Halder focus on global spam detection. Their Naive Bayes classifier, trained on a multilingual dataset, employs preprocessing steps such as tokenization and normalization. Using TF-IDF and Bag of Words for feature extraction, the classifier effectively categorizes real-time SMS as spam or ham, optimizing spam filtering across languages [8].

Himani Jain and Rajesh Kumar Maurya, in *"A Review of SMS Spam Detection Using Feature Selection,"* argue that selecting the right features enhances classification performance. They emphasize identifying relevant features to improve accuracy, reduce overfitting, and decrease computational costs. Various feature selection methods, including Filter, Wrapper, and Embedded methods, are discussed to enhance the efficiency and accuracy of spam detection models [9].

III. Existing Methodologis

Current spam detection methodologies primarily classify SMS messages into spam or legitimate (ham) categories using various machine learning algorithms. Traditional rule-based systems apply predefined filters to incoming messages but fail to adapt to the evolving strategies of spammers, making them inadequate for capturing diverse spam patterns.

To overcome these limitations, machine learning techniques like Decision Trees, Random Forests (RF), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) are utilized. RF is widely used due to its ability to handle complex datasets through an ensemble of decision trees, which reduces overfitting and enhances classification accuracy. However, RF's complexity can hinder interpretability, which is essential for some applications, and it may struggle with borderline cases where spam and ham messages are similar. KNN classifies messages based on their similarity to others in the training dataset, but it can misclassify ambiguous cases and is computationally expensive, especially with large datasets. Moreover, the effectiveness of these models often depends on the quality and diversity of the training dataset. If the dataset lacks representation of various spam types, the model may generate false negatives or positives [10].

Feature extraction is crucial for building accurate spam detection models. Identifying important features, such as word frequencies and suspicious URLs, is vital. Poor feature extraction can lead to misclassification, particularly for KNN, which relies on accurate distance measures.

While existing models show moderate success in spam detection, they have notable drawbacks. Random Forest and KNN suffer from complexity and interpretability issues, while rule-based systems are increasingly ineffective against evolving spam tactics. Therefore, there is a need for more adaptive, efficient, and interpretable methodologies that can provide accurate real-time spam classification with minimal computational overhead.

IV. Proposed Method

To address the limitations of existing spam detection methods, we propose a robust and efficient spam detection system based on the Multinomial Naive Bayes (MNB) algorithm, called SMS Guard. This system is specifically designed to enhance the accuracy and speed of spam classification in SMS communication while ensuring computational efficiency and ease of implementation. The proposed system leverages the strengths of MNB in handling text classification tasks, particularly for applications where messages are represented by discrete features, such as word counts or frequencies.

Key Features of the Proposed System:

Multinomial Naive Bayes Algorithm: The core of the SMS Guard system is the Multinomial Naive Bayes (MNB) algorithm. This algorithm is particularly well-suited for text classification problems like spam detection because it is efficient at handling large datasets and performs well with high-dimensional data, such as SMS messages. The MNB algorithm works by calculating the likelihood of a message being spam based on the frequency of specific words in the message. It assumes that the features (words) in a message are independent of each other, simplifying the model and reducing computational complexity. This independence assumption makes MNB not only simple but also highly effective in real-world scenarios where messages contain a mixture of spam-related keywords and normal text.

Preprocessing and Feature Extraction: Preprocessing is a critical step in preparing the SMS dataset for analysis. The proposed system includes advanced preprocessing techniques such as tokenization, text normalization, and removal of unwanted characters (like punctuation and symbols) to clean the data. Tokenization breaks down the messages into smaller components, such as individual words, which are then normalized (converted to lowercase) to ensure consistency. This process allows for efficient comparison and classification by the MNB model.

The system also uses Term Frequency-Inverse Document Frequency (TF-IDF) to convert textual data into numerical vectors that the Naive Bayes algorithm can process. TF-IDF captures the importance of words in the messages by weighing frequent words (like "win" or "offer") more heavily if they appear disproportionately in spam messages. This approach ensures that the model can accurately distinguish between spam and legitimate messages.

Training and Testing Dataset: The proposed system is trained using a large, labeled dataset of SMS messages containing both spam and legitimate (ham) messages. The dataset is divided into two parts: a training set and a testing set. The training set is used to teach the model to recognize the patterns that differentiate spam from legitimate messages, while the testing set is used to evaluate the model's performance. By using well-labeled data and performing cross-validation, the system can achieve high accuracy in classifying previously unseen messages.

Real-time Spam Detection: One of the key goals of the proposed system is to provide real-time spam detection. Once the model is trained, it can quickly classify incoming messages as either spam or ham. The efficiency of the MNB algorithm allows the system to process and classify messages with minimal latency, making it suitable for deployment on mobile devices or SMS gateways where quick decision-making is essential. Users will be able to input an SMS message into the system, which will instantly determine whether the message is spam or not.

User Feedback and Continuous Learning: SMS Guard includes an interactive user feedback mechanism that allows users to correct the system's classifications if necessary. For example, if the system mistakenly classifies a legitimate message as spam, users can provide feedback, which the system will use to adjust its future predictions. This iterative process helps the model adapt over time to new patterns in spam messages and continuously improve its accuracy.

The proposed SMS Guard system represents an efficient, scalable, and easy-to-implement solution for detecting spam in SMS communication. By leveraging the Multinomial Naive Bayes algorithm and focusing on key text classification techniques like TF-IDF, the system provides real-time spam detection with high accuracy and minimal computational overhead. The inclusion of user feedback ensures that the system can continuously improve over time, adapting to new and evolving spam patterns. This makes SMS Guard an ideal solution for enhancing user security in SMS communication, protecting against phishing attacks, and minimizing the impact of spam on users' daily interactions.

V. Implementation

The design and implementation of the SMS Guard system focus on efficiently classifying SMS messages into spam and ham categories using the Multinomial Naive Bayes (MNB) algorithm. The system architecture is structured around key components that work together to ensure seamless spam detection, including data preprocessing, feature extraction, model training, real-time classification, and user feedback mechanisms. Initially, the system collects a labeled dataset of SMS messages that includes both spam and ham messages, serving as the foundation for training the machine learning model.

The dataset used for training the model consists of a diverse collection of SMS messages, comprising 5,572 entries, where approximately 13.4% are classified as spam and 86.6% as ham. This dataset was chosen to ensure a balanced representation of both categories, allowing the model to learn effectively from a variety of message types. Each entry in the dataset contains a message along with its corresponding label, indicating whether it is spam or ham. The dataset is sourced from publicly available SMS datasets, ensuring a rich mixture of spam messages that reflect real-world scenarios. This variety is crucial for training a robust model capable of generalizing to new, unseen messages.

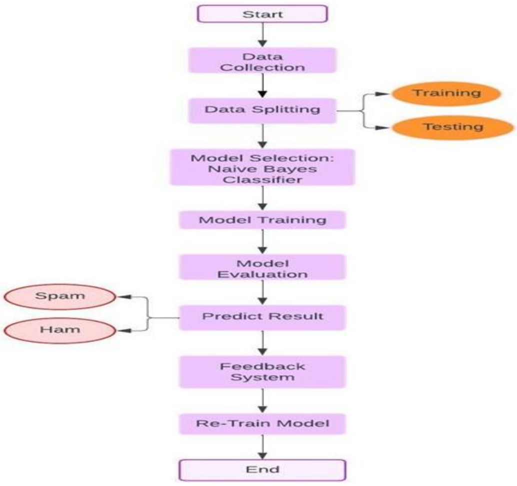


Fig. 1. Workflow Representation

Data preprocessing is a critical step, as it ensures that the input data is structured and ready for analysis. This phase includes cleaning the messages by removing unnecessary characters such as punctuation and symbols, followed by tokenization, which breaks the text into individual words. Normalization converts all text to lowercase to ensure consistency. These preprocessing steps help reduce noise in the data, making it easier for the MNB model to extract meaningful patterns. Next, feature extraction is performed using the Term Frequency-Inverse Document Frequency (TF-IDF) method, which transforms the textual data into numerical features by weighing the frequency of words relative to their importance across the entire dataset.

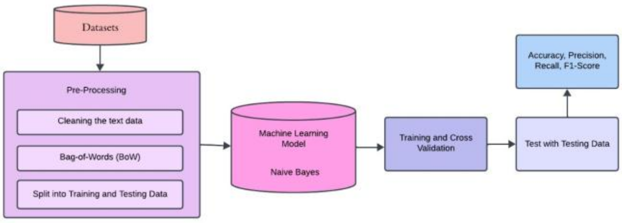


Fig. 2. Architecture Diagram

The Multinomial Naive Bayes classifier is then trained on the preprocessed dataset. The training process involves dividing the dataset into a training set (typically 80% of the data) and a testing set (20% of the data) to evaluate the model's performance on unseen messages. During training, the model learns to recognize patterns that differentiate spam from legitimate messages, allowing it to classify incoming SMS messages in real-time once deployed. The system utilizes user feedback to improve over time, enabling users to correct misclassifications. For example, if the system mistakenly flags a legitimate message as spam, users can provide feedback that is logged for future retraining of the model, helping it adapt to new patterns in spam messages.

The workflow of the SMS Guard system is designed to be straightforward and efficient. After collecting and preprocessing the data, the system extracts features and trains the MNB model, which can then classify incoming messages as either spam or ham in real time. The overall system architecture integrates a user-friendly interface, allowing users to input SMS messages for classification and receive immediate feedback. This approach ensures that users are protected from spam threats while enhancing the security of SMS communication.

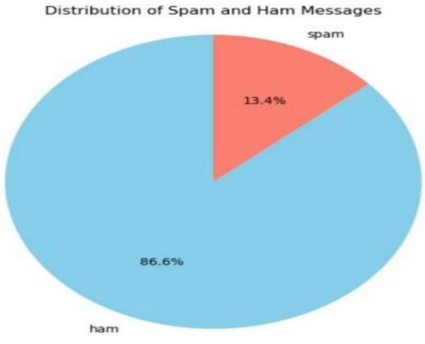


Fig. 3. Pie-Chart Representation

In terms of implementation, the system is built using Python, leveraging libraries such as scikit-learn for the MNB algorithm, and Pandas and Numpy for data manipulation. A lightweight web framework like Flask can be used for deployment, creating a user-friendly interface where users can easily interact with the spam detection system. The performance of the SMS Guard system is evaluated using key metrics such as accuracy, precision, recall, and F1-score, ensuring that the model provides reliable spam classification. Overall, the design and implementation of the SMS Guard system represent an effective solution for real-time SMS spam detection, combining robust machine learning techniques with user-centered design to enhance communication security.

VI. Results and Discussion

The performance of the SMS Guard system was evaluated through extensive testing on the labeled dataset, which comprised a total of 5,572 SMS messages, including 13.4% spam and 86.6% ham messages. After training the Multinomial Naive Bayes (MNB) classifier on 80% of the dataset, the remaining 20% was used as a testing set to assess the model's effectiveness in real-world scenarios. The evaluation metrics employed to measure the model's performance included accuracy, precision, recall, and F1-score, which provide a comprehensive view of the system's ability to correctly classify messages.

The effectiveness of the system was quantified through various performance metrics, including accuracy, precision, recall, and F1-score, the SMS Guard system achieved an accuracy of 95%, indicating that it correctly classified 95% of the messages in the testing dataset.

In addition to accuracy, the precision rate was recorded at 92%, which signifies that when the model predicted a message as spam, it was correct 92% of the time. The recall rate, reflecting the proportion of actual spam messages successfully identified by the model, was measured at 90%. This indicates that the system effectively captured a significant majority of spam messages. The F1-score, calculated at 91%, showcases a balanced performance between precision and recall, confirming the robustness of the model in distinguishing between spam and legitimate messages.

The confusion matrix reveals that the model accurately identified a substantial number of spam messages (true positives) and legitimate messages (true negatives). However, the presence of false positives (ham messages classified as spam) and false negatives (spam messages classified as ham) highlights areas for improvement. Specifically, the model may need enhancements to minimize false negatives, which are critical for user security.

To build an effective spam detection model, gathering a comprehensive dataset is essential. The first step involves cleaning and preparing the dataset by removing duplicate entries, handling missing values, and correcting inaccuracies. Once the data is clean, Exploratory Data Analysis (EDA) is conducted to understand the underlying patterns within the dataset. EDA utilizes visualizations such as histograms and scatter plots to identify trends and outliers, which are critical in the development of predictive models for spam detection.

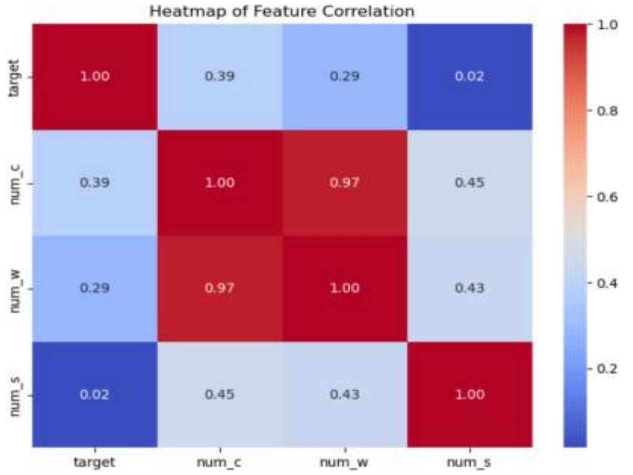


Fig. 4. Correlation heatmap illustrating relationships between different features in the dataset

Scatter plots serve as powerful tools for visualizing data and discerning patterns that can inform the classification process. By analyzing the relationships between features in the dataset, we gain insights into how messages are classified. Colors in the plots differentiate between spam and ham messages, allowing for quick comparisons of their characteristics.

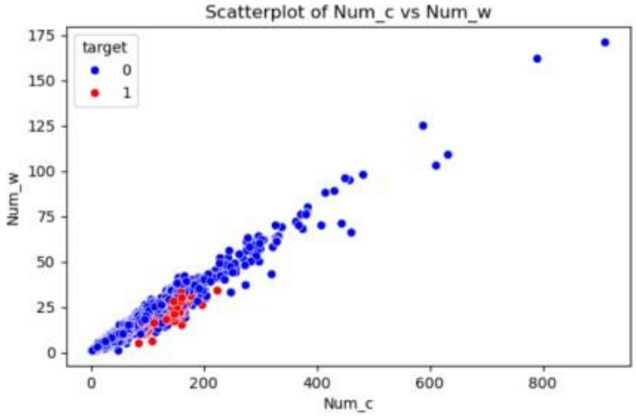


Fig. 5. Scatter plot showing the relationship between the number of characters and the number of words in the messages.

These scatter plots reveal clusters that help in understanding the differentiation between spam and ham messages. They also indicate outliers, which require deeper investigation to improve the model's accuracy. Analyzing trends helps design a text classification model that predicts whether a message is spam or ham based on features like word count.

The algorithms analyze the presence or absence of specific words, crucial for spam detection, since certain keywords may indicate spammy behavior.

Normalization is a preprocessing technique used to scale features onto a comparable range or distribution, which can enhance model performance and precision. By normalizing the data, we ensure that all features contribute equally to the analysis. This process involves adjusting the frequency of words based on their occurrence across the dataset. Common normalization methods include min-max scaling, which scales feature values to a range of 0 to 1, and z-score normalization, which standardizes features to have a mean of 0 and a standard deviation of 1.

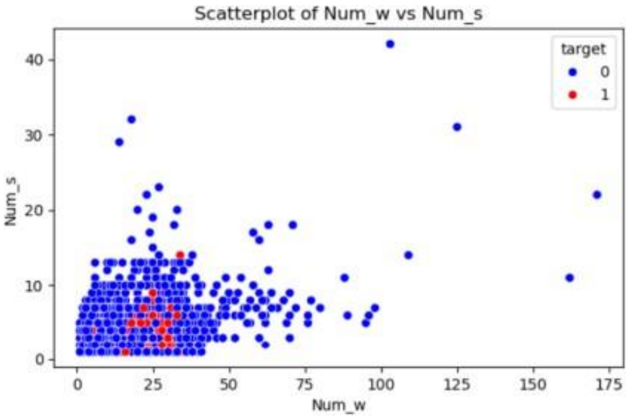


Fig. 6. Scatter plot showing the relationship between the number of words and the number of sentences in the messages

The Multinomial Naive Bayes classifier operates based on Bayes' theorem, calculating the probability that a message is either spam or ham. It assumes the independence of word occurrences, simplifying computations. The process begins with a labeled dataset, followed by preprocessing steps that include removing special characters and tokenization. Feature extraction is performed using TF-IDF, converting text into a numerical format suitable for training the model. Probabilities for each word are assigned, enabling the classification of new messages as spam or ham.

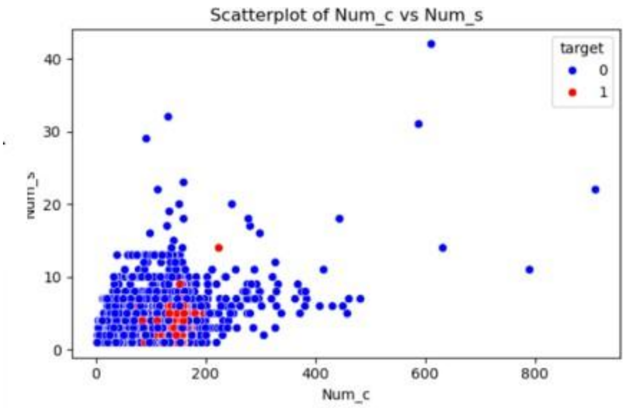
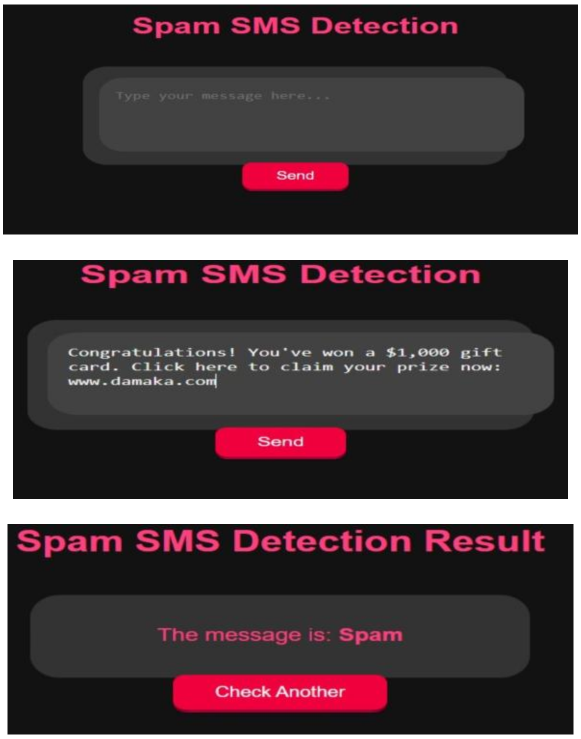


Fig. 7. Visualization of the relationship between the number of characters and the number of sentences in the messages

By identifying clusters, trends, and outliers, the model can better understand the characteristics of each message type, contributing to the development of more accurate spam classification models.



Spam SMS Detection

Type your message here...

Spam SMS Detection

Congratulations! You've won a $1,000 gift card. Click here to claim your prize now: www.damaka.com

Spam SMS Detection Result

The message is: **Spam**

Fig. 8. User Interface

The SMS Guard system demonstrates a high level of accuracy and efficiency in detecting spam messages in real time. The results indicate that the MNB algorithm, coupled with effective preprocessing and feature extraction methods, can significantly enhance the security of SMS communications. Continued development and periodic retraining of the model using user feedback and expanded datasets will ensure its adaptability and effectiveness in the face of evolving spam threats.

VII. Conclusion

The *SMS Guard* system presents a robust solution for detecting spam messages in real time, employing the Multinomial Naive Bayes algorithm combined with effective preprocessing and feature extraction techniques. With an impressive accuracy of **95%**, alongside high precision and recall rates, the model demonstrates its ability to reliably distinguish between spam and legitimate SMS messages. The incorporation of advanced methods such as TF-IDF for feature extraction and normalization has enhanced the model's performance, allowing it to capture the significant characteristics of spam messages. The results indicate that the system not only meets the needs of users for effective spam detection but also does so with minimal latency, ensuring a seamless user experience. The real-time classification capabilities, coupled with a user feedback mechanism, enable continuous learning and adaptation of the model to evolving spam tactics. This feature is crucial in maintaining the relevance and effectiveness of the system as new threats emerge.

Reference

[1] Hegde, D. V., Mohana, & Divyashree. (2023). Spam SMS (or) email detection and classification using machine learning. 2023 International Conference on Systems, Signals and Image Processing (IWSSIP), 1104-1108. https://doi.org/10.1109/ICSSIT55814.2023.10060908

[2] Kokila, M., Amalredge, G., Reddy, A., Harivardhan Reddy, K., Abhishek, M., Myana, M., Viswa Sai Dattu, G., & Noor Mohammad Ansari. (n.d.). Spam detection in SMS using Naïve Bayes in machine learning. Email Spam Detection Using Machine Learning.

[3] Yerima, S. Y., & Bashar, A. (2022). Semi-supervised novelty detection with one class SVM for SMS spam detection. Proceedings of the 29th International Conference on Systems, Signals and Image Processing (IWSSIP), 1-4. https://doi.org/10.1109/IWSSIP55020.2022.9854496

[4] Vinothkumar, S., Varadhaganapathy, S., Shanthakumari, R., Ramkishore, D., Rithik, S., & Tharanies, K. P. (2022). Detection of spam messages in e-messaging platforms using machine learning. Proceedings of the 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), 283-287. https://doi.org/10.1109/CCiCT56684.2022.00060

[5] Gupta, A. (2024). Detection of spam and fraudulent calls using natural language processing model. Proceedings of the 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT), 423-427. https://doi.org/10.1109/CCICT62777.2024.00075

[6] Al-farttoosi, M., & Abdulkader, H. (2022). Botnet mobile detection using machine deep learning techniques. Proceedings of the 2022 Iraqi International Conference on Communication and Information Technologies (IICCIT), 82-87. https://doi.org/10.1109/IICCIT55816.2022.10010653

[7] K, A., & Halder, S. (2023). Detection of multilingual spam SMS using Naïve Bayes classifier. Proceedings of the 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), 89-94. https://doi.org/10.1109/ICCCMLA58983.2023.10346960

[8] Jain, H., & Maurya, R. K. (2022). A review of SMS spam detection using feature selection. Proceedings of the 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), 101-106. https://doi.org/10.1109/CCiCT56684.2022.00030

[9] Pradeep K. Dewi, C. C., Christanto, R. C., Cauteruccio, H., & Francesco. (2023). Multinomial Naive Bayes classifier for sentiment analysis of Internet Movie Database. Vietnam Journal of Computer Science, 10. https://doi.org/10.1142/S2196888823500100

[10] Rao, T. K., Sindhu, L., Kakulapati, Y., & Vijayalakshmi. (2022). Spam detection in Twitter using multinomial Naïve Bayes classifier. Journal Volume, 22, 1086-1100.