International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Enhanced SMS Spam Classification Using Machine Learning: A Voting Classifier Approach

Sukhvinder Singh Bamber
University Institute of Engineering & Technology, Panjab University SSG Regional Centre, Hoshiarpur, Punjab, India

Abstract **— The short message service (SMS) gained popularity after it was first offered as a service in the second-generation terrestrial mobile network structure. Its popularity has been utilized by some advertising agencies and others to distribute unwanted advertisements, convey advertising proposals, and send unwanted content to the end users. These unwanted messages, also referred to as spam, prevent the users from getting the wanted messages and cause them frustration and annoyance. Spam messages are increasingly common on mobile devices, not only causing nuisance but also exposing users to serious security risks. Many spam messages are used to deceive recipients into providing personal information through phishing schemes. This study proposes a machine learning-driven solution for identifying spam messages by leveraging Natural Language Processing (NLP) techniques alongside classifiers like naïve bayes and various ensemble algorithms. These models achieve high accuracy and precision through TF-IDF vectorization and a strategic combination of classifiers, achieving an accuracy of 98.16% and a precision of 99.17%. This model is deployed in a web-based interface, enabling users to detect spam in real-time and thereby enhancing mobile communication security. This study highlights the importance of combining robust feature extraction techniques with ensemble learning to address challenges like dataset imbalance and the brevity of SMS messages. Future work will focus on optimizing the model for mobile platforms and expanding its applicability to multilingual and regional spam detection scenarios.**

Keywords **— Spam SMS, Machine Learning, TF-IDF, Naïve Bayes, Ensemble Learning, Real-time Detection.**

I. Introduction

Technological progress has significantly simplified and accelerated daily life, enabling cost savings. Mobile phones, with their extensive features, have revolutionized communication, making it more accessible. Text messaging, in particular, offers an affordable and straightforward way for people to stay in touch without needing an internet connection. However, this ease of communication has been misused by some individuals to distribute bulk spam messages to random users, often for malicious purposes or personal advantage. Short Message Service, or text messaging, is a form of communication that now plays a key role in the lives of billions of people around the world.

Unfortunately, the widespread use of SMS has also made it a major target for unsolicited message senders whom the authors commonly refer to as spammers. These spammers often use SMS as their main medium to spread their unsolicited messages, or spam, thanks to the ease of use, speed, and low cost of unauthorized SMS. Nowadays, the act of sending unsolicited messages is a significant problem because it constitutes a growing proportion of fraudulent activities and apparently disturbs the SMS user. Currently, the widespread use of SMS in everyday communication has resulted in SMS receiving an excessive number of unsolicited messages or spam. This has led to the rise of SMS advertising. In this paper, the authors have proposed some ways to detect and prevent spam, but users of mobile services must remain aware of the potential detriments to themselves and their devices from using such services.

In recent years, text messaging has become a vital communication tool for individuals and businesses alike. With this growth, however, comes an increase in unsolicited spam messages, ranging from advertisements to potentially harmful phishing attempts. Spam SMS not only overcrowds message inboxes but also presents significant security risks, as numerous messages are crafted to deceive individuals into sharing confidential personal or financial details. This scenario underscores an urgent need for reliable solutions that can automatically recognize and filter spam messages effectively. According to a survey, 76% of Indians receive at least three spam or promotional text messages per day. The survey found that:

- 15% of respondents receive 1–2 unwanted SMS per day
- 33% of respondents receive 3–5 unwanted SMS per day
- 18% of respondents receive 6–10 SMS per day
- 25% of respondents receive more than 10 SMS per day

SMS spam detection has been a well-researched domain, with numerous techniques achieving commendable accuracy rates. For example, Random Forest algorithm, combined with chi-square feature selection, achieved an impressive accuracy of 97.50%. Other innovative approaches, such as danger theory paired with krill herd optimization, also showed good performance with an accuracy of 96%.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Using Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction along with the Random Forest classifier produced a similar accuracy of 97.50%. Neural network-based models have also been used effectively, reaching an accuracy of 97.67% in classifying spam and legitimate messages. In contrast, the Latent Dirichlet Allocation (LDA) algorithm, while less accurate, was able to detect SMS spam with 90.40% accuracy.

These methods demonstrate the variety of machine learning and deep learning techniques used to tackle SMS spam detection, each offering unique contributions to solving the problem. However, challenges persist, particularly due to the imbalance in datasets, where legitimate (ham) messages vastly outnumber spam messages. This imbalance can impact model performance, requiring strong techniques that can effectively handle unseen data. Moreover, the brevity of SMS messages often limits the availability of meaningful features, making the feature extraction process more challenging.

To overcome these challenges, ensemble learning has emerged as a promising solution. By combining multiple models, ensemble methods enhance overall accuracy and robustness, addressing some of the key issues in SMS spam detection. Several traditional Machine learning classifiers, including Naïve Bayes and Support Vector Machines (SVM), have also proven effective, particularly when complemented with ensemble techniques.

Naïve Bayes continues to be a widely used method for text classification because of its simplicity and dependability. Both traditional and modern approaches show great potential in addressing the ongoing challenge of SMS spam detection. However, recent research highlights that ensemble techniques, such as Voting and Stacking Classifiers, can enhance accuracy by leveraging the strengths of multiple models. These methods increase robustness by enabling the classifier to make better predictions, considering the variety in SMS messages.

This study contributes in two significant ways. Firstly, it showcases the effectiveness of an ensemble-based model, which surpasses traditional single classifiers in accuracy and precision. Secondly, it stresses the importance of real-world deployment, allowing users to identify spam messages in real-time. By integrating advanced machine learning techniques with a user-friendly interface, this work offers a practical and reliable solution for detecting spam in mobile communications.

II. Page Layout

Research on spam detection has evolved significantly over the past two decades, transitioning from rule-based systems to machine learning models capable of learning from data. Initial spam filters relied on simple rules and keywords to flag spam messages. However, as spammers adapted their tactics, these approaches became less effective, prompting researchers to seek more adaptable solutions.

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have been applied to categorize SMS messages as either spam or legitimate, achieving an accuracy of 97.50%. In addition, hybrid models combining Convolutional Neural Networks (CNN) and LSTM achieved accuracies of 97% and 95%, respectively. Another noteworthy approach involved using Hidden Markov Models (HMM), which successfully classified and filtered SMS spam with a 95.90% accuracy.

Authors of [1] Dharani et al. in 2023 proposed a model based on Naïve Bayes algorithm and term frequency-inverse document frequency vectorizer and trained the proposed model with the kaggle dataset for avoiding being scammed by the scammers. Researchers of [2] in 2021 proposed a modified transformer model for detecting SMS spam messages. Then evaluated the proposed model on SMS Spam Collection v.1 dataset and UtkM1's Twitter Spam Detection Competition dataset. The proposed model achieved promising results indicating it's future use for the similar problems. Akbari and Sajedi [3] developed a GentleBoost algorithm for SMS spam detection in binary and unbalanced data. Authors identified that in majority of cases especially binary and unbalanced data GentleBoost performed better as compared to other boosting classifiers. Further, authors have suggested the approach of extracting word attributes by removing the unused features and optimizing them. Manju et al. [4] in 2023 carried out the work in the field of SMS spam filtering after analyzing that SMS spam has seen an exponential increase in the past few years leading to the loss of data in pile of spam messages. Authors have proposed features which gives high accuracy when classifying SMS messages.

Researchers Tuan et al. [5] in 2023 presented various methods for detecting SMS spam in Vietnamese language. Authors carried out the extensive experimentation based on algorithms like: SVM, Naïve Bayes, Random Forests, CNN and LSTM.

Outcomes proved that LSTM and CNN performed much better than the traditional machine learning models. Asmitha and Kavitha [6] in 2024 investigated how NLP techniques can be used to detect spam messages. Researchers further proposed a model examined for generating disaster tweets. The proposed system's performance was evaluated using parameters like: Precision, Recall, F-score, FNR and FPR. In 2017, Abdulhamid et al. [7] provided the extensive review of available methods, challenges and future directions for spam detection, filtering and mitigation of mobile SMS spams. The main objective of this review was to assist the researchers to identify open areas that needed improvements.

Hosseinpur and Shakibian [8] in 2023 have investigated that unbalanced proportion of spam, ham data and the extraction of efficient features from short messages are the main challenges in SMS spam detection problem. Authors have proposed an ensemble learning method based on random forest and logistic regression algorithms to increase the accuracy of SMS spam detection. Bajaj et al. [9] in 2023 analyzed the security weaknesses of smishing detection method by employing advanced attack techniques to generate adversarial text. Authors further carried out the comparative analysis of transformer models like: BERT and DistilBERT on the basis of Attack Success Rates (ASR).

Aparna and Halder [10] in 2023 focused on the detection of multilingual spam SMS using the Naive Bayes classifier. The proposed approach leverages the Naive Bayes algorithm's simplicity and efficiency to classify SMS messages as spam or non-spam, while specifically addressing the multilingual aspect. In 2024, researcher Guppa [11] developed a user privacy and security appeal based on NLP. This ensures that discussions remain completely anonymous by not collecting, storing or recording user data that resolves data protection issues with many programs for SMS and SMS filtering. Authors implemented special approvals on Call-E and run them in the background without interfering with calls or messages. Also protected users' mobile experiences like a quiet sentinel who is always willing to act. Users can receive real-time security through Call-E-NLP-based technology, which has successfully recognized and blocked spam and deceptive information.

Siji et al. [12] in 2023, Spam Ham Categorization (Spam Cat Model) presented a solution that distinguishes between spam and ham messages and uses established text classification methods.

This innovative approach focuses on the analysis of language characteristics, idiomatic representations, and content patterns to significantly improve the accuracy of spam recognition. By fighting the security and user experience aspects, the SPAM-CAT model provides a paradigm shift in SMS communication management. Researcher Salman et al., 2024. [13] highlighted the current landscape challenges of SMS-SPAM recognition and filtering. To address these challenges, the authors presented new SMS data records. This includes over 68 KSS messages, including 61% legitimate SMS (HAM) and 39% spam messages. To characterize the data records, researchers conducted a longitudinal analysis of the evolution of spam, then extracted semantic and syntactic features, and evaluated and compared the performance of well-known SMS-SPAM detection methods for machine learning, a range of flat machine learning approaches. The researchers investigated the robustness of existing SMS-SPAM recognition models and the robustness of common spam services to spammer beneficial techniques.

Qazi et al. [14] in 2024, checked the existing cutting edge methods of three groups: spam reviews, individual spammers, and group spam. The authors categorized machine learning (ML) and deep learning (DL) technologies for spam detection, showing a total of 10 metrics. This made accuracy the most frequently used in the detection of spam detection with accuracy of 24% of studies and 22% of studies (25%) in relation to ML-based techniques. Furthermore, the metrics for F measurements showed that the area under the curve (AUC) and F1 score assessment metrics increased the use of Amazon datasets by 7%. The proposed study concludes that most SMS spam filter strategies are the main solutions. Furthermore, the taxonomy of the most modern methods has been developed, and the conclusion has been drawn that a considerable number of studies are using these existing SMS-Anti spam applications. Author: Nayak et al. [15] In 2024, machine learning algorithms need to be constantly improved to maintain the newly developed spam techniques and provide a safe online environment. The proposed study used Kaggle data records originally intended for spam recognition. Data required some conversion and migration to perform multilingual spam recognition in French, German and English. Intensive preparations such as stopword removal, tokenization, and categorization according to the language provide greater flexibility in data records to examine complex spam patterns in multilingual settings. Various machine learning algorithms such as multinomial NB, XGBoost, LSTM, BERT, and more were used accordingly to achieve the desired result.

Kabbi et al. [16] 2023 proposed an SMS spam recognition approach consisting of preprocessing, distinctive extraction, characteristic fusion, feature selection, and classification. The proposed model was developed to simultaneously record local, time, and global SMS features using a hybrid deep learning model to improve the representation of characteristics. The authors evaluated the proposed model using UCI data records and used cross-validation to ensure robustness of the results compared to traditional deep learning algorithms such as HF and BERT. In 2023, Yerima and Bashar [17] proposed a detection system for SMS SPAM with a semi-monitoring novelty recognition approach, based on the class SVM classifier. This system is created as an abnormality detector that only learns from normal SMS messages. This means that the identification model can be implemented without the example of marked SMS spam training. The authors evaluated the proposed system with a benchmark dataset consisting of 747 SMS spam and 4827 spam messages. The results show that the proposed method surpasses traditional approaches to machine learning based on binary, frequency, or TF-IDF-Wörser approaches. Agarwal et al. [18] by adding Indian messages to global SMS data records, were inspired to tackle tasks and filter mobile phones as ham or spam for Indian users. Authors analyzed various classifiers for machine learning on a large body of SMS messages for the Indian population. Authors Gandhi et al. [19] analyzed various SMS spam recognition methods and their efficiency in the detection of spam messages. Data were examined using three machine learning models LSTM, DNN, and BI-LSTM models. The authors found that the proposed method using data records from SMS text messages can recognize spam messages with high accuracy and highly effective. Of the three models, DNN produced the highest accuracy of 95.6522%, followed by BI-LSTM with accuracy of 93.9799% and LSTM with 92.9766%.

Researcher Qian et al. [20] published that text messages from short messages are essential, but there is a serious problem with spam. The authors proposed a service-solution in which data mining is used by graphs to recognize spammers as non-pampers without checking the content of messages, and to distinguish and recognize spam. In 2017, Lota and Hossian [21] conducted a systematic literature overview of SMS-SPAM detection technology. To this end, the authors looked at the research work available from 2006 to 2016 and selected 17 articles for the proposed study, checking the methods used, approaches, algorithms, their advantages and disadvantages, rating scales, discussions on data recording, and ultimately the results.

Sethi et al. [22] analyzed and examined the relative strength of various algorithms for machine learning to recognize spam messages sent on mobile devices. The authors recorded Open Public Dataset data and created two data records for testing and validation purposes. The accuracy of spam message detection was a top priority in ranking these algorithms. The results clearly showed that different algorithms of machine learning under different characteristics tend to classify different ways in spam messages. In 2020, authors Julis M.R. and Alagesan S. [23] dealt with the obligation to organize various messages as ham or spam for Indian users by adding Indian messages to the overall data record of accessible SMS data records.

[24] Alzahrani A. et al. Researcher of. In 2019, when filtering spam messages, techniques such as logistic regression, naive Bayes algorithm, support vector machines (SVMs), neural networks measure the accuracy of determining the most effective method for filtering spam messages. Based on the results, neural networks are best executed as a trained classification form model, with incoming messages being classified as ham or spam. Nisar N. et al. [25] Traditional machine learner implementations of machine learning such as Naive Bayes (NB), Support Vector Machine (SVM), k-nearest neightre (k-nn), decision tress (dt), Random Forest (RF), and adaboost draw individually and individually to record accuracy and other performance measurements. Furthermore, the researcher's ensemble speech classifier used the results of each individual classifier used and calculated the outcome classes with a large number of voices. Author of son D. T. et al. in 2022. [26] proposed the use of deep learning techniques to recognize botnet networks. The authors analyzed that perceptions of botnet networks are becoming increasingly complex and that traditional methods are no longer effective. Therefore, one of the recent urgent issues is finding effective solutions for bot network detection based on deep learning characteristics such as scalability, performance, execution time, and interpretability.

The literature survey indicates that traditional classifiers such as Naïve Bayes and SVM are effective for spam detection, while ensemble methods like Random Forest and AdaBoost provide added robustness. This research builds on these findings by applying TF-IDF vectorization and ensemble methods to create a model that balances accuracy with computational efficiency for SMS spam detection.

III. Proposed Methodology

The process of detecting spam messages involves several key stages, including data cleaning, data exploration and analysis, data preparation, algorithm training, and final deployment through the PyCharm IDE as shown in fig.1.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Each of these stages is crucial for accurately classifying messages as either spam or legitimate (ham). The steps are explained in detail:

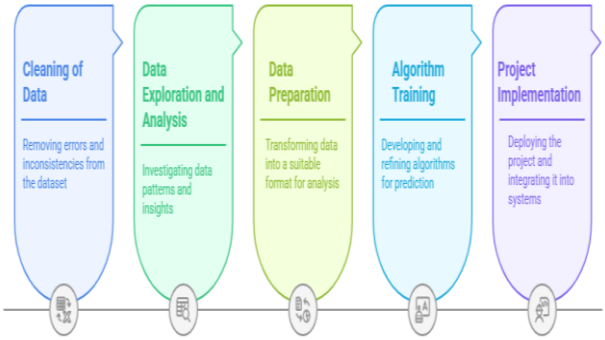

Figure 1: Block Diagram for SMS Spam Detection

a) Cleaning of Data involved removing irrelevant columns from the dataset and assigning labels, where spam messages were denoted as 1 and ham messages as 0. The dataset was then checked for missing values and duplicates. Any duplicate records were dropped to ensure data quality and consistency before proceeding to the next stage.

b) Exploratory Data Analysis (EDA) focused on understanding the distribution of spam and ham messages. The proportion of spam and ham messages was visualized using a pie chart. Additionally, three new columns were created to count the number of characters, words, and sentences in each message. Spam and ham messages were analyzed separately, with insights derived from the differences in their character, word, and sentence counts. Correlation analysis was performed to understand the relationships among these features.

c) Data preparation played a vital role in getting the text ready for machine learning models. All text was standardized to lowercase to maintain consistency across the dataset. The text was broken down into individual words, and unnecessary elements like stop words, punctuation, and special characters were removed to reduce noise. Words were then simplified to their root forms through stemming, making the data more manageable. A Word Cloud visualization was created to showcase the most common terms in spam and ham messages, offering valuable insights into their linguistic characteristics.

d) Model Building process started with the vectorization of text using TF-IDF, a technique that weighs words based on their frequency and importance within the dataset. The data was divided into training and testing sets, and several models were trained, including different variations of Naïve Bayes classifiers, such as Gaussian, Multinomial, and Bernoulli. These models were assessed based on their accuracy and precision to identify the best performing approach for classifying spam messages.

For deployment, the trained TF-IDF vectorizer and classifier were saved as vectorizer.pkl and model.pkl using the pickle library. An interactive web application was then developed using **Streamlit**, a Python framework, which provided a simple and intuitive interface. The application features an input box where users can type or paste a message and a "Predict" button to classify the message as spam or legitimate (ham). Once deployed, the application generates a local URL that allows users to interact with the model in real time.

The workflow begins by categorizing the dataset into two types: spam messages (marked as 1) and legitimate messages (marked as 0).Each message is then broken down into its individual components, such as characters, words, and sentences. The next step involves preparing the data by converting text to lowercase, performing tokenization, and applying stemming to simplify the words. After preparing the data, it is vectorized using the Naïve Bayes algorithm, marking the completion of the algorithm training phase. Finally, the model is implemented and deployed, ensuring the accurate classification of messages based on the learned features as shown in figure 2.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

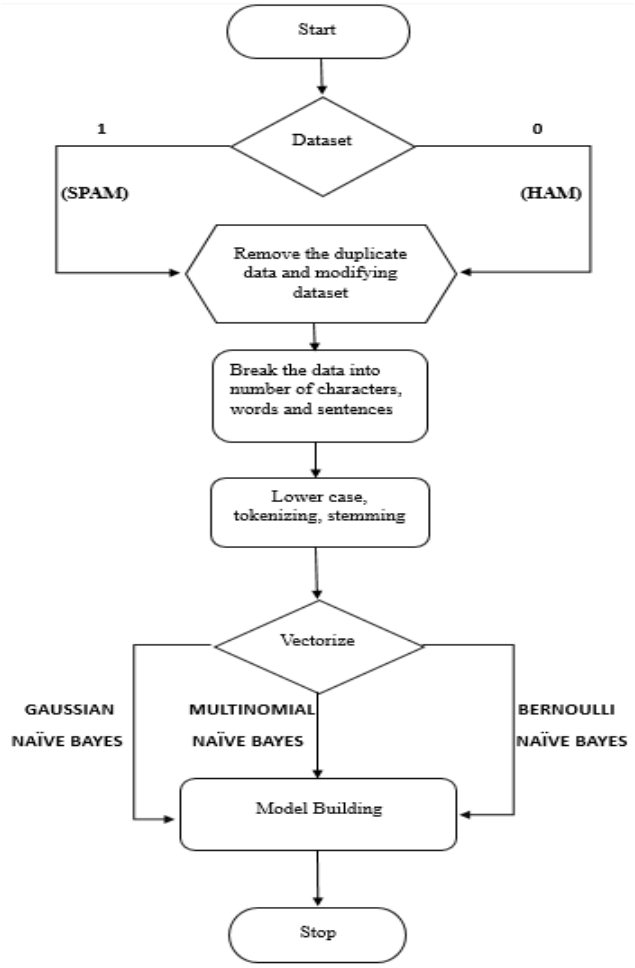

Figure 2: SMS Spam Detection Flowchart

The development process involved Jupyter Notebook, an open-source, web-based platform that supports a variety of programming languages. Jupyter was used for tasks like data exploration, visualization, analysis, and the creation of documents. Python libraries required for the project were installed and imported directly into the Jupyter Notebook environment, providing a streamlined workflow for iterative development.

A. Algorithm

Algorithm 1 describes an end-to-end method for training and deploying an SMS spam filtering model. Labeled SMS messages (ham and spam) first go through data preprocessing that includes lowercasing, tokenization, stop word removal, and stemming. The preprocessed text is then converted to TF-IDF vectors using the top 3,000 features.

Algorithm 1: Building and deploying the SMS Spam Detection Model

Input:
Labeled SMS messages (spam and ham).

Output:
A trained model that classifies new messages as spam or ham.

Data Preprocessing
Convert text to lowercase, tokenize, remove stop words, and apply stemming.

TF-IDF Vectorization
Transform preprocessed text into TF-IDF vectors, selecting the top 3,000 features.

Model Training
Split the dataset into training and test sets (80%-20%).

Train various classifiers, including Naïve Bayes, SVM, and ensemble methods.

Evaluate each model on accuracy and precision.

Model Selection and Evaluation
Identify the best-performing model based on accuracy, precision, and computational efficiency.

Deployment
Save the model and vectorizer as .pkl files.

For new messages: preprocess, vectorize, and classify in real time.

Then, data is divided into test and training sets (80%-20%), and different classifiers such as Naïve Bayes, SVM, and ensemble algorithms are trained and tested on the basis of accuracy and precision. The model that performs the best is chosen to be deployed, and both the model and vectorizer are saved for real-time spam classification.

B. Time Complexity Analysis

Time complexity measures how the time required for an algorithm grows as the input size increases. Researchers have implemented Big-O notation, which shows the worst-case scenario for algorithm growth as follows:

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

$$T(n) = C * f(n) \quad (1)$$

where:

T(n): Total time complexity.

C: A constant representing time per operation (it doesn't scale with input size).

f(n): A function that represents how the number of operations grows with input size n.

Common growth rates describe how the runtime or complexity of an algorithm increases with the input size.

- *Constant Time O(1):* Takes the same time regardless of input size. Example: Accessing an element in an array.
- *Logarithmic Time O(log n):* Reduces the problem size by a factor each step, like in binary search.
- *Linear Time O(n):* Directly proportional to input size n, like iterating over an array.
- *Quadratic Time O(n^2):* Happens with nested loops, like matrix operations.

These growth rates helps in evaluating algorithm efficiency and scalability in computational tasks and system design. Naïve Bayes algorithm has been implemented for calculating probabilities in all n samples. The time complexity is:

$$T(n, m) = O(n * m) \quad (2)$$

Here, n is the number of samples, and m is the number of features. This makes Naïve Bayes efficient, especially for large datasets.

Support Vector Machines works by finding the optimal decision boundary, which can be computationally intensive. For a linear kernel, time complexity is:

$$T(n, m) = O(n^2 * m) \quad (3)$$

For a non-linear kernel (e.g.: RBF), time complexity increases to:

$$T(n, m) = O(n^3) \quad (4)$$

Random Forest (RF) consists of t decision trees. Each tree has a complexity of $O(n * \log n)$, where n is the number of samples. So, for t trees, the overall time complexity is:

$$T(n, m, t) = O(t * n * \log n) \quad (5)$$

TF-IDF Vectorization method calculates the term frequency-inverse document frequency for n documents and m terms. Its time complexity is:

$$T(n, m) = O(n * m) \quad (6)$$

Ensemble Classifiers (Voting and Stacking) methods combine predictions from multiple base models. If there are k classifiers, each with a complexity $O(n * m)$, the total time complexity becomes:

$$T(n, m, k) = O(k * n * m) \quad (7)$$

All have different temporal complexity depending on input size and model parameters. SVMs are computationally demanding, especially when using nonlinear kernels. Random Forest scales with trees and samples, but TF-IDF and ensemble algorithms scale with document and classifier numbers.

C. *Dataset Description*

The dataset for this study was obtained from Kaggle and consists of 5,570 SMS messages, categorized into spam and non-spam (ham). Analysis revealed that 12.63% of the messages were spam, while 87.37% were non-spam, reflecting a typical imbalance found in real-world communication. During the data cleaning process, duplicate and unlabeled messages were removed, and irrelevant columns were dropped to streamline the dataset. A pie chart, shown in figure 3, was used to visualize the proportions of spam and non-spam messages, providing a clear representation of the dataset's structure.

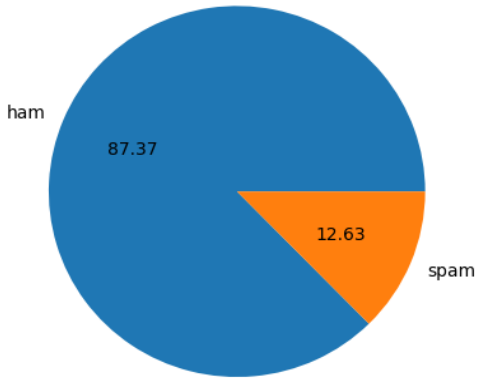

Figure 3. Spam and Ham Classification Dataset

To ensure data quality, further checks were performed to eliminate inconsistencies and anomalies. Deeper exploration revealed that spam messages were generally longer, containing more characters, words, and sentences compared to non-spam messages. This analysis helped in uncovering linguistic patterns that informed the feature engineering and model-building processes, ensuring the dataset was well-prepared for effective training and testing of the spam detection system.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

IV. Results And Discussions

To create an SMS spam detection model, the dataset is cleaned to remove any duplicate and unlabeled messages. Next, exploratory data analysis (EDA) is performed to examine various parameters such as the number of characters, words, and sentences. During data preprocessing, the text is converted, tokenized, removed unnecessary elements and applied stemming. Once the data is preprocessed, model building is done by vectorizing it. After that, the necessary files are imported into PyCharm and construct the model to achieve the desired results. The dataset consists of 87.37% ham (non-spam) messages and 12.63% spam messages.

A. Text Characteristics Analysis

Analyzing the textual characteristics of SMS messages offers valuable insights into the distinguishing features of spam and non-spam messages. In this study, three primary characteristics were examined for each message: the number of characters, the number of words, and the number of sentences. The total count of characters provided a basic measure of message length, while the number of words was determined using tokenization to capture the overall message structure. Additionally, the number of sentences was calculated to understand the complexity and formatting of each message.

These metrics were analyzed for all 5,169 messages in the dataset, enabling a detailed comparison of spam and non-spam messages. A summary of these characteristics is presented in table 1, offering a clear overview of the dataset's linguistic properties and their relevance to spam detection.

Table 1: Statistical Summary of Text Characteristics

Metric	Mean	Std. Dev.	Min	25th Percentile	50th Percentile	75th Percentile	Max
Number of Characters	78.98	58.24	2	36	60	117	910
Number of Words	18.46	13.32	1	9	15	26	220
Number of Sentences	1.97	1.45	1	1	1	2	38

As shown in Table 1, the average SMS message contains approximately 78 characters and 18 words, with an average of around 2 sentences. Notably, the maximum number of characters and words in a single message are 910 and 220, respectively, suggesting that spam messages may often be longer than typical ham messages.

B. Comparative Analysis of HAM and SPAM Messages

To better understand the structural differences between spam and ham (non-spam) messages, the authors conducted a comparative analysis of three primary characteristics: number of characters, words and sentences. This analysis highlights significant distinctions between spam and non-spam messages, providing insights that can help improve spam classification.

C. Descriptive Statistics

The descriptive statistics for each characteristic are presented in table 2. Notably, spam messages tend to be longer, containing more characters, words, and sentences on average compared to ham messages.

Table 2: Comparative Summary Statistics for Ham and Spam Messages

Metric	Ham (Non-Spam)	Spam
Count	4,516	653
Mean Characters	40.46	124.89
Mean Words	17.12	27.63
Mean Sentences	1.82	2.97
Max Characters	910	224
Max Words	220	46
Max Sentences	38	9

Spam messages are generally longer in terms of character and word count, with an average of approximately 138 characters and 28 words, compared to 70 characters and 17 words for ham messages. The greater length and structure (i.e., number of sentences) of spam messages may be attributed to promotional content, which often includes detailed information.

D. Visual Analysis of Message Characteristics

To visualize these differences, the authors plotted histograms for the number of characters and words, comparing ham and spam messages (figure 4 and figure 5).

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

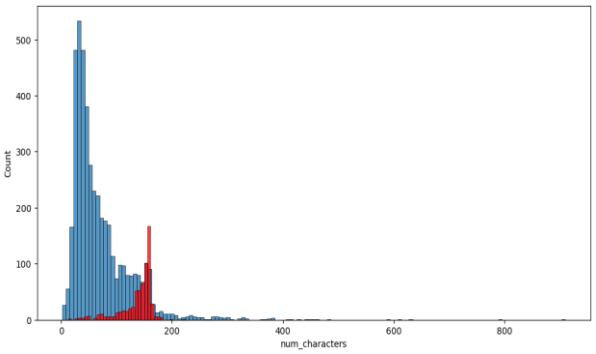

Figure 4: Distribution of Number of Characters in Ham and Spam Messages

Figure 4 demonstrates that spam messages typically have higher character counts, peaking around 130-160 characters, while ham messages are more evenly distributed across lower character counts. This difference suggests that character count could serve as an effective feature for distinguishing between spam and ham messages.

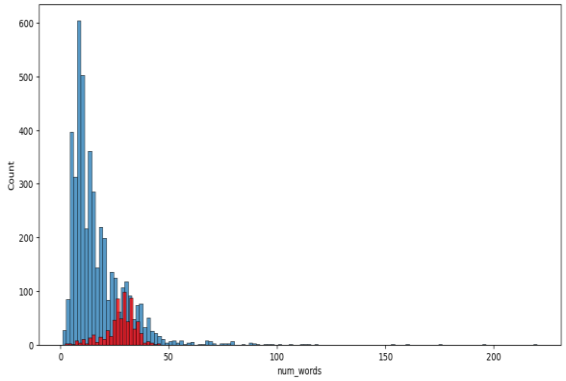

Figure 5: Distribution of Number of Words in Ham and Spam Messages

As shown in figure 5, spam messages tend to contain more words, with a clear peak around 25-35 words. In contrast, ham messages have a broader distribution across fewer words, peaking around 10-20 words. This further supports the observation that spam messages are generally longer and more content-rich, which may be due to the promotional or informative nature of spam content.

E. Feature Correlation Analysis

The pair plot in figure 6 illustrates the relationships between numerical features, segmented by the target variable (ham or spam). Each scatter plot in the matrix represents the relationship between two features, color-coded based on whether the message is spam or ham.

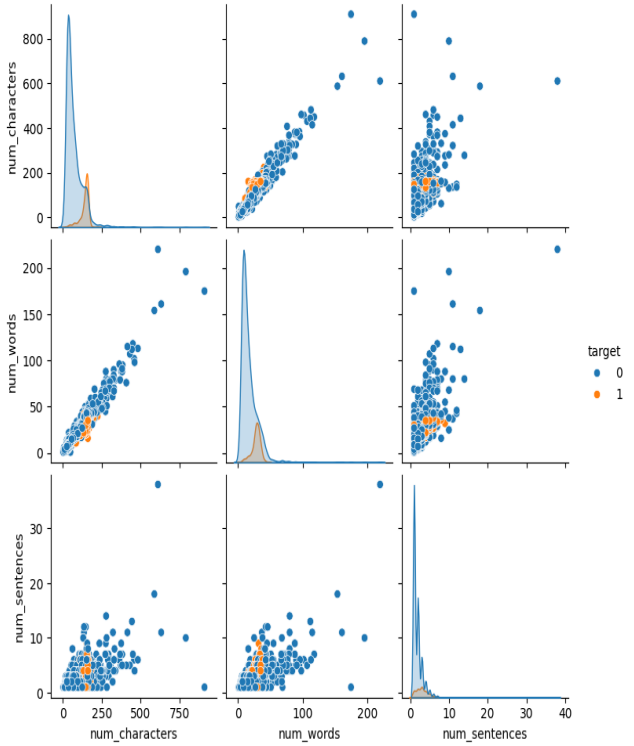

Figure 6: Pair Plot of Numerical Features by Target Class

Figure 6 provides insights into how feature distributions vary between ham and spam messages. For example, spam messages generally have a higher number of characters and words, as seen in the scattered separation of points. The pair plot serves as a visual confirmation that spam messages tend to be longer, with more words and sentences.

Figure 7 includes a heatmap that illustrates the correlation between the number of characters, words, and sentences, with 1 representing spam and 0 representing ham.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

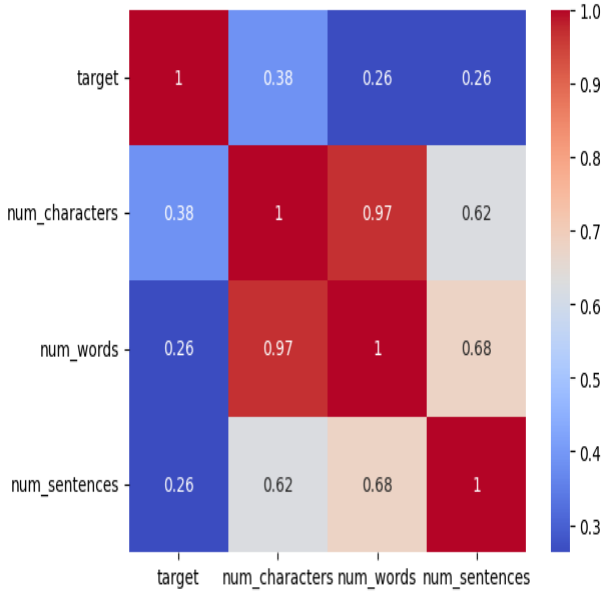

Figure 7: Heat map showing correlation

In figure 7, the authors observe the correlation matrix for the numeric features. The number of characters and number of words exhibit a high positive correlation, suggesting that longer messages tend to have more words. However, number of sentences shows a moderate correlation with both characters and words, indicating that sentence structure may vary more between ham and spam messages.

To visualize the most common words in spam and ham messages, the authors generated word clouds for each category. These word clouds highlight the frequency of words in the dataset, providing a clear view of the typical vocabulary associated with spam and ham messages.

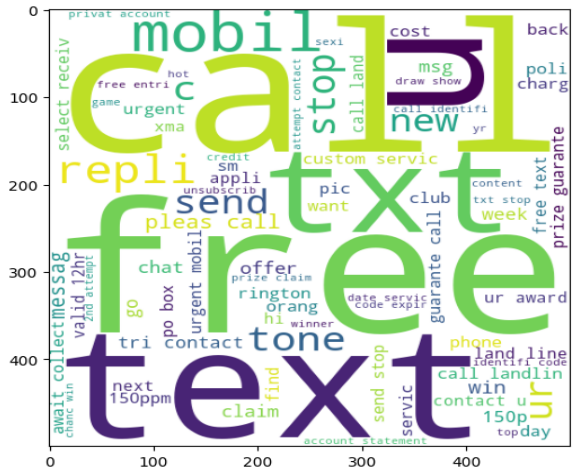

Figure 8: Word Cloud of Spam Messages

Figure 9: Word Cloud of Ham Messages

Figures 8 and 9 show the most frequent words in spam and ham messages, respectively. In the spam word cloud (Figure 8), prominent terms include “free,” “win,” and “urgent,” reflecting the common vocabulary used in spam messages to attract attention. In contrast, the ham word cloud (Figure 9) shows common conversational words, indicating a more informal tone in non-spam messages. These visualizations provide valuable insights into the language patterns that may help distinguish spam from non-spam messages.

F. Performance of Various Algorithms Tested for Model

This analysis helped determine which algorithm would be most suitable and yield the best results. Accuracy measures how close the results are to the true or known value, while precision measures how close the results are to one another. Both metrics are essential for tracking and reporting results. Figure 10 illustrates a bar graph showcasing the accuracy and precision scores achieved by different algorithms during the model-building phase, including K-Nearest Neighbors (KN), Naïve Bayes, Random Forest, Support Vector Classifier (SVC), Extra Trees Classifier (ETC), and Linear Regression.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

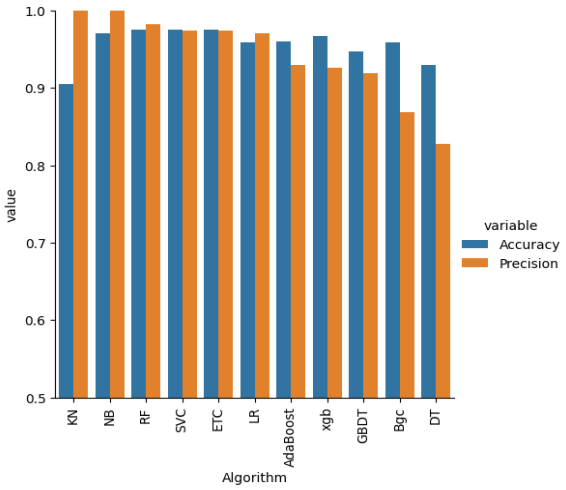

Figure 10: Performance Evaluation and Model Comparison of Algorithms

After comparing the performance of each algorithm, Naïve Bayes was chosen as it outperformed the others in terms of both accuracy and precision. RF, SVC, and Extra Trees Classifier work extremely well with both measures above 95%. AdaBoost, XGBoost, and Gradient Boosted Decision Trees algorithms are slightly lower in precision than in accuracy. DT and Bagging Classifier have the worst performance of the models tested. Ensemble and tree-based models generally have excellent results but slightly fall behind in precision from simpler models. Performance metrics for each model, including accuracy and precision, are summarized in table 3.

Table 3: Results of performance metrics for each model, including accuracy and precision

Model	Accuracy (%)	Precision (%)
Multinomial Naïve Bayes	97.10	100.00
Random Forest	97.58	98.29
Voting Classifier	98.16	99.17
Stacking Classifier	97.97	93.98

Voting Classifier had the best accuracy of 98.16% and high precision of 99.17%, surpassing other classifiers. Random Forest had well-balanced performance with accuracy of 97.58% and precision of 98.29%.

Multinomial Naïve Bayes had perfect precision but low accuracy of 97.10%. Stacking Classifier had good precision (97.97%) but low precision (93.98%).

Files exported from Jupyter are uploaded into PyCharm, where the execution of code generates a local host URL. This interface includes a text field where users can input an SMS to determine if it is spam, along with a "Predict" button to perform the classification.

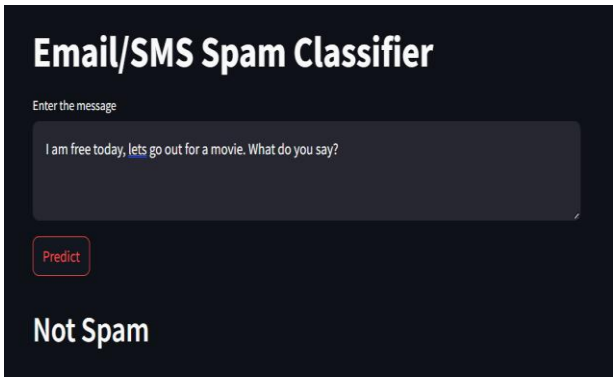

Figure 11: Result Displaying Ham SMS

Figure 11 showcases the outcome for a legitimate (ham) SMS. Upon clicking the "predict" button, the system provides a clear indication of whether the given SMS is classified as spam or not. This process is simple and intuitive, ensuring that users can easily understand and obtain accurate results. Moreover, this approach can be seamlessly adapted for integration into mobile applications, making it versatile for various platforms while maintaining its reliability and efficiency.

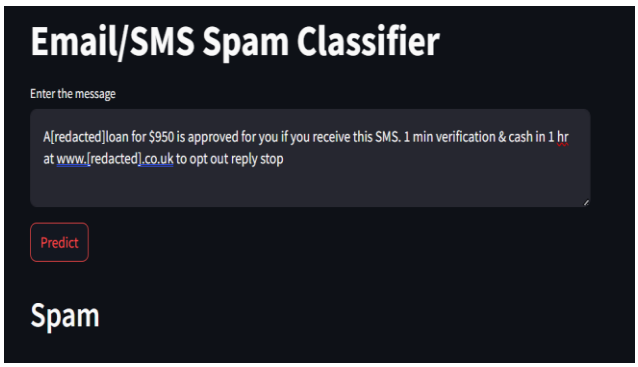

Figure 12: Result Displaying a Spam SMS Classification

Figure 12 displays the result of a spam SMS detection process. In the "Enter the Message" field, users can paste a received SMS to assess whether it is spam or legitimate (ham).

This method is easy to use, providing a reliable and efficient way for users to quickly determine the nature of their messages.

The proposed model can also be effectively applied to email spam detection. However, due to the longer and more detailed nature of emails compared to SMS messages, a larger and more comprehensive dataset is required. Detecting email spam may actually be more straightforward than detecting SMS spam for several reasons. First, the longer text allows for clearer identification of suspicious patterns and intentions. Additionally, the availability of sender information, such as email addresses, can aid in detecting spam, as many spam emails often originate from known sources or addresses with suspicious characteristics. Moreover, users are generally more aware of email spam, which helps in identifying potential threats. By leveraging these factors, email spam detection can be streamlined and more accurate, benefiting from both the content and context of the message.

V. Conclusion And Future Scope

This study demonstrates that combining TF-IDF vectorization with machine learning classifiers, particularly ensemble methods, offers an effective approach to SMS spam detection. The Voting Classifier achieved the best performance, with an accuracy of 98.16% and a precision of 99.17%, underscoring the value of ensemble techniques for handling diverse and evolving spam content. This research shows that advanced machine-learning approaches can provide reliable and robust spam filtering solutions for mobile communication. Despite these promising results, there are areas for improvement. Future work could explore the use of semantic-rich NLP techniques, such as word embeddings and transformer models, to capture deeper contextual cues in spam messages. Expanding the dataset to include multilingual and region-specific spam would enhance the model's adaptability. Optimizations for mobile environments, like model pruning and on-device learning, could further improve deployment feasibility. Overall, this study offers a solid foundation for future advancements in spam detection, contributing to safer and more secure mobile communication.

REFERENCES

[1] Dharani V., Hegde D. and Mohana (2023), "Spam SMS (or) Email Detection and Classification using Machine Learning", 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Bengaluru, Karnataka, India, 2023, doi: 10.1109/ICSSIT55814.2023.10060908.

[2] Liu X., Lu H. and Nayak A. (2021), "A Spam Transformer Model for SMS Spam Detection," IEEE Access, Vol. 9, May 2021, DOI: 10.1109/ACCESS.2021.3081479.

[3] Akbari F. and Sajedi H. (2015), "SMS Spam Detection using Selected Text Features and Boosting Classifiers", 7th International Conference on Information and Knowledge Technology (IKT2015), Urmai, Iran, doi: 10.1109/ICSSIT55814.2023.10060908.

[4] Manju G., Nivethika D., Mondal S. J. and Sthapak H. (2023), "SMS Spam Detection and Filtering of Transliterated Messages", Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), Chennai, India, doi: 10.1109/ICCEBS58601.2023.10449289.

[5] Tuan V. M., Duong D. T. and Anh T. Q. (2023), "Proposing Appropriate SMS Spam Detection Approaches for Variations of the Vietnamese Language", International Conference on Computing and Communication Technologies (RIVF), Hanoi, Vietnam, 2023, doi: 10.1109/RIVF60135.2023.10471859.

[6] Asmitha M. and Kavitha C.R. (2024), "Exploration of Automatic Spam/Ham Message Classifier Using NLP", 9th International Conference for Convergence in Technology (I2CT), IEEE, Pune, India, 2024, doi: 10.1109/I2CT61223.2024.10544236.

[7] Abdulhamid S. M., Latif M. S. A., Chiroma H., Osho O., Salaam G. A., Abubakar A.I. and Herawan T. (2017),"A Review on Mobile SMS Spam Filtering Techniques", IEEE Access, Vol. 5, pp. 26768-26778, doi: 10.1109/ACCESS.2017.2666785.

[8] Hosseinpour S. and Shakibian H. (2023), "An Ensemble Learning Approach for SMS Spam Detection", 9th International Conference on Web Research (ICWR), Tehran, Iran, 2023, doi: 10.1109/ICWR57742.2023.10139070.

[9] Bajaj A. and Vishwakarma D. K. (2023), "Deceiving Deep Learning-Based Fraud SMS Detection Models through Adversarial Attacks", 17th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, Bangkok, Thailand, 2023, doi: 10.1109/SITIS61268.2023.00059.

[10] Aparna A. K. and Halder S. (2023), "Detection of Multilingual Spam SMS Using Naïve Bayes Classifier", IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Hamburg, Germany, doi: 10.1109/ICCCMLA58983.2023.1034696.

[11] Gupta A. (2024), "Detection of Spam and Fraudulent Calls Using Natural Language Processing Model", 6th International Conference on Computational Intelligence and Communication Technologies (CCICT), Chandigarh, India, doi: 10.1109/CCICT62777.2024.00075.

[12] Siji S. R., Prakash B., Karpurapu R. and Mohan A. S. (2023), "Hierarchical Classification Model for SMS: An Evolving Model for HAM Categorization", 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Benguluru, India, doi: 10.1109/ICIMIA60377.2023.10425985.

[13] Salman M., Ikram M. and Kaafar M. A. (2024), "Investigating Evasive Techniques in SMS Spam Filtering: A Comparative Analysis of Machine Learning Models", IEEE Access, Vol. 12, doi: 10.1109/ACCESS.2024.3364671.

[14] Qazi A., Hasan N., Mao R., Abo M. E. M., Dey S. K. and Hardaker G. (2024), "Machine Learning-Based Opinion Spam Detection: A Systematic Literature Review", IEEE Access, Vol. 12, doi: 10.1109/ACCESS.2024.3399264.

[15] Nayak A., Kumari R., Pal D., Jana S., Bhardwaj A. and Dasude P. M. (2024), "Multilingual SMS Spam Detection Using BERT and LSTM", International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, doi: 10.1109/ICICET59348.2024.10616322.

[16] Kabbi H. A. A, Derakhshi M.R. F. and Pashazadeh S. (2023), "Multi-Type Feature Extraction and Early Fusion Framework for SMS Spam Detection", IEEE Access, Vol. 11, doi: 10.1109/ACCESS.2023.3081479.

[17] Yerima S. Y. and Bashar A. (2022), "Semi-Supervised Novelty Detection with One-Class SVM for SMS Spam Detection", 29th International Conference on Systems, Signals, and Image Processing (IWSSIP), Sofia, Bulgaria, doi: 10.1109/IWSSIP55020.2022.9854496.

[18] Agarwal S., Kaur S. and Garhwal S. (2015), "SMS Spam Detection for Indian Messages", 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, doi:10.1109/NGCT.2015.7375198.

[19] Gandhi C., Samangi P. K., Saxena M. and Sahoo A. K. (2023), "SMS Spam Detection Using Deep Learning Techniques: A Comparative Analysis of DNN Vs LSTM Vs Bi-LSTM", International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), IEEE, Greater Noida, India, doi: 10.1109/CISES58720.2023.10183634.

[20] Xu Q., Xiang E. W., Yang Q., Du J. and Zhong J. (2012), "SMS Spam Detection Using Noncontent Features", IEEE Intelligent Systems, Vol. 27, Issue. 6, doi: 10.1109/IWSSIP55020.2022.9854496.

[21] Lota L. N. and Hossian B. M. M. (2017), "A Systematic Literature Review on SMS Spam Detection Techniques", International Journal of Information Technology and Computer Science, MECS, Vol. 7, pp. 42-50.

[22] Sethi P. Bhandari V. and Kohli B. (2017), "SMS Spam Detection and Comparison of Various Machine Learning Algorithms", IEEE International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), Gurgaon, India, pp. 28-31.

[23] Julis M. R. and Alagesan S. (2020), "Spam Detection in SMS Using Machine Learning Through Text Mining", International Journal Scientific and Technology Research, Vol. 9, No. 2, pp. 498 - 503.

[24] Alzahrani A. and Rawat D.B. (2019), "Comparative Study of Machine Learning Algorithms for SMS Spam Detection", IEEE SoutheastCon 2019, Huntsville, AL, USA, pp. 1-6.

[25] Nisar N., Rakesh N. and Chhabra M (2021), "Voting-Ensemble Classification for Email Spam Detection", IEEE International Conference on Communication Information and Computing Technology (ICCICT), Mumbai, India, pp. 1-6.

[26] Son D. T., Tram N.T.K. and Hieu P.M. (2022), "Deep Learning Techniques to Detect Botnet", Journal of Science and Technology on Information Security, Vol. 1, No. 15, pp. 85-91.