A Research paper of Spam SMS Classification

Vaishnavi Shrikant Pataskar
Computer Science and Engineering Department
Parul Institute of Technology
vaishnavispataskar@gmail.com

Abstract

The Spam SMS Classification project leverages data science techniques to develop an effective system for identifying and filtering spam messages in SMS communications. With the increasing prevalence of spam texts, which often contain fraudulent content, advertisements, or phishing attempts, it is crucial to implement reliable classification methods. This project employs machine learning algorithms and natural language processing to analyze and categorize SMS messages as either spam or not spam. The methodology includes data collection from publicly available datasets, data preprocessing to clean and prepare the text for analysis, and feature extraction. Various machine learning models, including Naive Bayes, Support Vector Machines (SVM), and Random Forests, are trained and evaluated for their accuracy and effectiveness in classifying messages. The project aims to achieve a high level of accuracy while minimizing false positives and negatives, thereby enhancing user experience and communication safety. By providing a user-friendly interface for real-time classification, the project addresses a significant challenge in mobile messaging today. Ultimately, this work contributes to the ongoing efforts in the field of spam detection and aims to adapt continuously to evolving spam tactics.

Keywords— Spam SMS Classification; Machine Learning; Natural Language Processing; Naive Bayes; Support Vector Machine; Random Forest; Data Pre-processing; Feature Extraction

1. Introduction

Spam SMS messages are unwanted texts that people receive on their phones. These messages can be annoying and sometimes even harmful, as they often contain advertisements, scams, or attempts to trick users into giving away personal information. With so many people using mobile messaging today, it's important to find effective ways to identify and block these spam messages. [3]

Many traditional methods for dealing with spam, like manually blocking numbers or ignoring unwanted texts, are not very effective. Spammers are constantly changing their tactics, making it hard for users to tell which messages are real and which are fake. [2]

This project focuses on creating a system that can automatically classify SMS messages as spam or not spam using data science techniques. We will use machine learning, which is a way for computers to learn from data, and natural language processing, which helps computers understand human language. [4]

The project involves several steps:

1. Data Collection: We will gather a dataset of SMS messages that are labeled as spam or not spam.
2. Data Pre-processing: We will clean the text messages to make them easier to analyze.
3. Feature Extraction: We will convert the text into numerical data that machine learning models can understand.
4. Model Training: We will use different machine learning algorithms to train the model to recognize spam messages.
5. Evaluation: We will test how well the model works to ensure it can accurately classify messages.

2. Literature Review

Research has shown that various techniques can be used for spam detection, including:

Camponovo G and Cerutti D [4] Various researchers have explored different machine learning techniques to effectively identify and filter out spam messages.

Fu J, Lin P, Lee S. [6] explained early work primarily focused on traditional algorithms, such as Naive Bayes and decision trees, which showed promising results but had limitations in handling large and complex datasets.

Cleff E.B [5] explained recent studies have shifted towards more advanced methods, including Support Vector Machines (SVM) and ensemble techniques like Random Forests, which have demonstrated higher accuracy and robustness in classifying spam. Additionally, the use of natural language processing (NLP) techniques, such as text normalization and feature extraction methods.

A.K. Jain et al. [10] The literature on spam SMS classification explores various methods and techniques to effectively identify unwanted text messages. Many studies highlight the use of machine learning algorithms, such as Naive Bayes and Support Vector Machines, which have shown good results in distinguishing between spam and ham messages. Researchers often emphasize the importance of feature extraction methods, like TF-IDF and Bag of Words, which convert text into numerical data that algorithms can analyse.

M.A. Al-Ghamdi et al. [9] explained about some studies also explore deep learning approaches, such as recurrent neural networks (RNNs) and convolutional neural networks

(CNNs), which can capture complex patterns in SMS data. Additionally, many papers discuss the challenges of evolving spam tactics, noting that models need regular updates to maintain their effectiveness. Overall, the literature suggests that a combination of well-chosen algorithms and robust feature extraction techniques can significantly improve spam classification accuracy.

- Dataset: As Fig 1 shows, a popular dataset for this task is the SMS Spam Collection dataset, which contains labelled messages classified as spam or ham. This dataset is widely used in research and can be found on platforms like Kaggle and the UCI Machine Learning Repository [3].

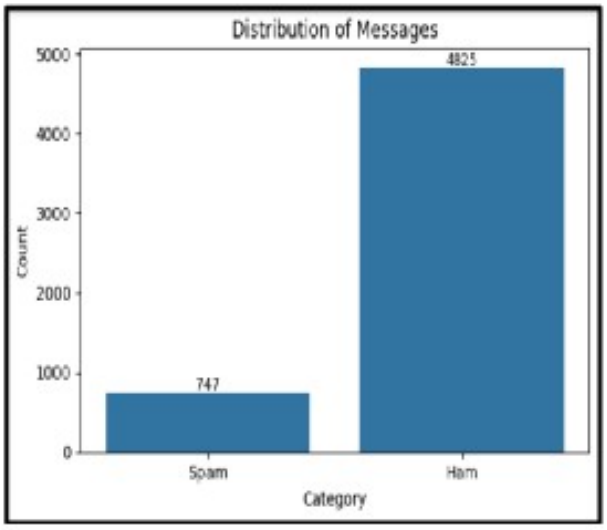

Figure 1: Distribution of Messages

- Data Pre-processing: Effective pre-processing techniques are crucial for improving model performance. Common steps include:
  - Converting text to lowercase.
  - Removing special characters, numbers, and stop words.
  - Tokenizing the text and applying stemming or lemmatization to reduce words to their base forms. [1]
- Feature Extraction: Transforming text data into numerical format is essential for machine learning models. Techniques such as:
  - TF-IDF (Term Frequency-Inverse Document Frequency): This method helps in identifying the importance of words in the messages.
  - Count Vectorization: Another approach that counts the occurrences of each word in the messages. [2]
- Model Selection: Various machine learning algorithms can be employed for classification, including:
  - Naive Bayes: Known for its effectiveness in text classification tasks.
  - Support Vector Machines (SVM): Useful for handling high-dimensional data.
  - Decision Trees and Random Forests: These can provide good accuracy and interpretability.[1][2]
- Model Evaluation: It is important to evaluate the model using metrics such as accuracy, precision, recall, and F1-score. Cross-validation techniques can also be applied to ensure the model's robustness. [2]
- Implementation: The final model can be implemented in a user-friendly application where users can input SMS messages for classification, helping them filter out spam effectively. [1]

3. Challenges, Objectives and Security

Several challenges exist in the spam SMS classification process:

1. Data Quality: Collecting a diverse dataset that represents various types of spam and legitimate messages can be difficult.
2. Evolving Spam Techniques: Spammers continuously change their tactics, making it hard for models to keep up.
3. False Positives and Negatives: The system must minimize mis-classifications, where legitimate messages are marked as spam or vice versa.
4. User Privacy: Ensuring that user data is handled securely and respectfully is crucial.

3.1. Objectives

The main objectives of the Spam SMS Classification project are:

1. Automating Message Filtering: Instead of relying on manual identification, the project leverages machine learning algorithms to classify messages in real time, saving users time and effort.
2. Adapting to Evolving Spam Techniques: The model can learn from new data, allowing it to stay effective against evolving spam tactics and trends.
3. Providing Insightful Data: The project can analyse spam trends, offering users and organizations valuable insights into the types of spam messages they encounter.

3.2. Security

Ensuring the security of the Spam SMS Classification project is very important. Here are some key aspects to consider:

1. Data Protection: We need to keep all user data safe. This means using encryption, which is a way to scramble data so that only authorized users can read it. By encrypting SMS messages and any personal information, we can protect it from hackers.

2. User Privacy: It's essential to respect user privacy. We should only collect the data we need and make sure we don't share any personal information with third parties without the user's consent.

3. Secure Access: We need to implement secure login methods to ensure that only authorized users can access the system. This might include using strong passwords, two-factor authentication, or other security measures to verify users' identities.

4. Proposed Work

The proposed work involves developing a machine learning model that uses a combination of traditional and modern algorithms to classify SMS messages. The system will be trained on a large dataset, ensuring it can adapt to new spam techniques over time.

The proposed work for spam SMS classification aims to create a system that can accurately identify unwanted messages (spam) and separate them from regular messages (ham). To start, we will collect a dataset of SMS messages that are already labelled as spam or ham. Next, we will clean and prepare the data by removing duplicates and unnecessary words, and then convert the messages into a numerical format that a machine learning model can understand.

We will try different models, such as Logistic Regression, Naive Bayes, and Support Vector Machines, to see which one works best for our task. After training the models on part of the data, we will test them on a separate set to evaluate their performance. We will also fine-tune the models to improve accuracy. Finally, we will create an easy-to-use application where users can input their messages and receive instant spam classification. The goal is to help users manage their SMS more effectively by filtering out unwanted messages.

The key steps in the project include:

1. Collect SMS data from available datasets.
2. Clean the data by removing unwanted characters and normalizing text.
3. Tokenize the text into individual words.
4. Extract features using techniques like TF-IDF.
5. Train the model using selected algorithms.
6. Evaluate the model using metrics like accuracy and precision.
7. Deploy the model with a user-friendly interface.

Fig 3 Shows the complete step-by-step workflow for creating an SMS spam classification system.

1. Raw SMS Data: The process starts with collecting text messages which includes legitimate messages and spam messages.
2. Data Cleaning: Any missing information or problematic data in the messages is handled, ensuring the dataset is complete and usable.

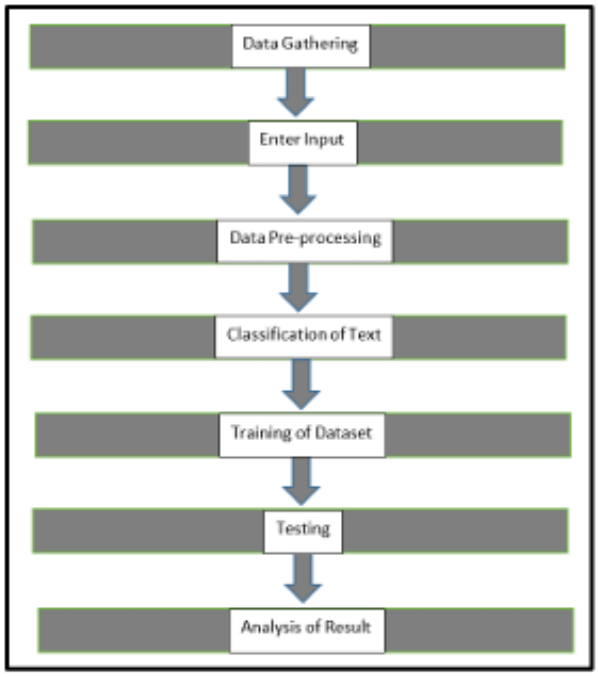

Figure 2: Spam SMS Classification Process

3. Label Encoding: The messages are labelled with numbers for processing - spam messages are assigned "0" and legitimate (ham) messages are assigned "1".
4. Text Pre-processing: The text messages are prepared by:
   - Converting all letters to lowercase.
   - Removing common words that don't help identify spam (stop words).
   - Reducing words to their base form.
5. Data Splitting: The dataset is divided into two parts:
   - 80% for training the model (what it learns from)
   - 20% for testing how well it works
6. TF-IDF Vectorization: Text messages are converted into numbers the computer can understand by:
   - Using a minimum of 1 occurrence for a word to be included
   - Focusing on English stop words
   - Setting lowercase=True to treat words the same regardless of capitalization
7. Logistic Regression Model: The mathematical algorithm that learns patterns in the data to distinguish between spam and legitimate messages.
8. Model Evaluation: Checking how well the model performs:
   - Training Accuracy: 96.77% (how well it learned from training data)

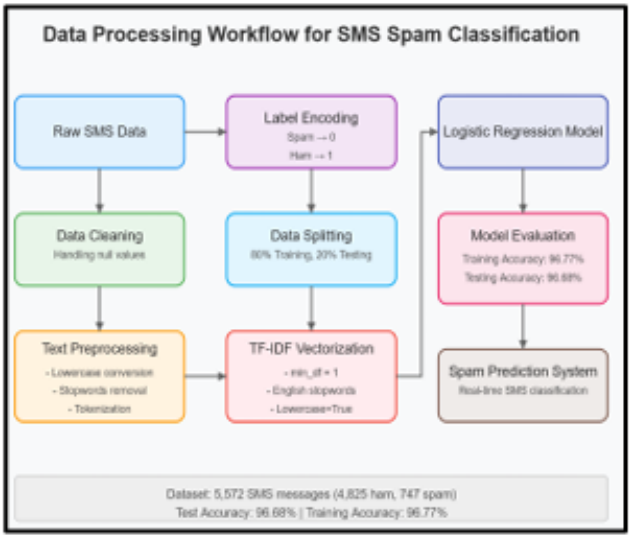

Figure 3: Spam SMS Classification - Data Processing Workflow

- Testing Accuracy: 96.68% (how well it performs on new data)

9. Spam Prediction System: The final working system that can classify incoming SMS messages in real-time.

The bottom of the diagram confirms the dataset details and the accuracy scores, showing that this approach is highly effective with over 96% accuracy for identifying spam messages.

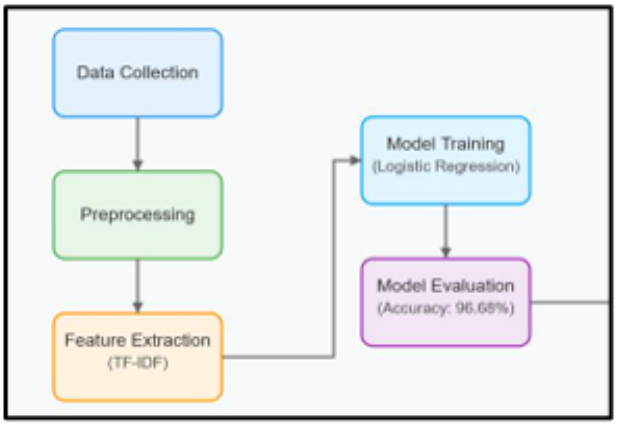

Figure 4: Spam SMS Classification - Architecture

Fig 4. Shows how a spam detection system for text messages works from start to finish:

1. Data Collection: First, the system gathers SMS messages - both spam and legitimate ones. This creates a dataset that the system can learn from.

2. Pre-processing: The raw text messages are cleaned up. This typically involves removing punctuation, converting text to lowercase, removing stop words (common words like ”the” or ”and”), and possibly stemming words (reducing them to their base form).

3. Feature Extraction (TF-IDF) (Orange Box): The cleaned text is converted into numerical features that a machine learning model can understand. TF-IDF (Term Frequency-Inverse Document Frequency) is used here, which measures how important specific words are in each message compared to the entire collection of messages. Words that appear frequently in spam but rarely in legitimate messages get higher weights.

4. Model Training (Logistic Regression) (Blue Box): The system uses these numerical features to train a Logistic Regression model. This algorithm learns patterns that distinguish spam from legitimate messages by assigning probabilities.

5. Model Evaluation (Accuracy: 96.68%) (Purple Box): The trained model is tested to see how well it performs. The accuracy of 96.68% means the model correctly identifies whether a message is spam or legitimate about 97 out of 100 times.

6. Deployment: After confirming the model works well, it’s implemented in a real-world environment where it can automatically classify incoming SMS messages as either spam or legitimate.

This architecture follows a standard machine learning pipeline where data flows from collection through various processing stages until a working model is deployed. The high accuracy indicates that Logistic Regression works effectively for this particular text classification task.

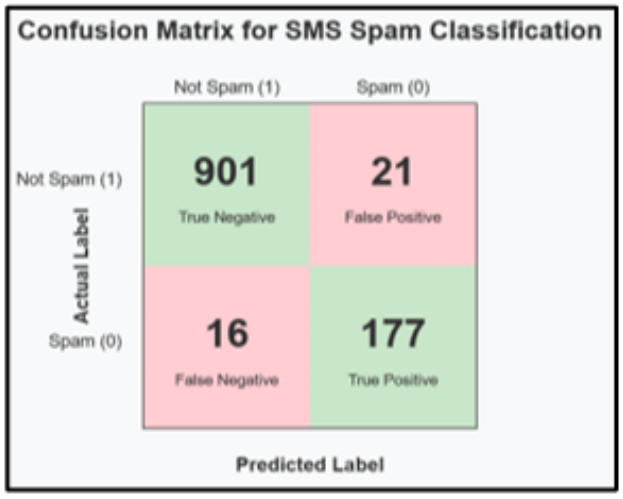

Figure 5: Spam SMS Classification – Confusion Matrix

Fig 5. Shows your model is performing well, with high accuracy. The low number of false negatives means users won’t receive many spam messages, while the relatively low number of false positives means legitimate messages rarely get blocked incorrectly.

5. Result Analysis with different Output and Comparison

The output generated from the spam SMS classification code indicates the performance of a logistic regression model

used to classify SMS messages as either spam or ham. In the output (Fig 6), we see predicted probabilities for various messages, where each entry includes the message index and its likelihood of being spam.

As shown in Table 1, the model achieved an accuracy of approximately 96.8% on the training data and 96.7% on the test data, suggesting that it performs well in identifying spam messages while generalizing effectively to new data. The final prediction confirms that the input message is classified as spam, demonstrating the model's capability to detect unwanted text.

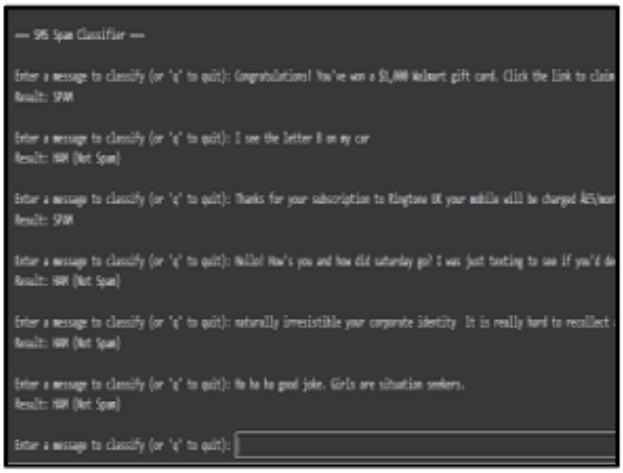

Figure 6: Output

Fig 6. Shows the result of spam SMS. In the provided block named Enter a message to classify (or 'q' to quit), paste the SMS message which is received to check whether the message is spam or ham.

5.1. Analysis of different Output

Figure 7: Spam SMS Classifier – SPAM

Fig 7. Shows the result of SPAM SMS. Upon clicking Enter button it shows provided SMS is spam or not spam. This is a process to be followed to check whether the message is spam or not spam. The process is fairly comfortable with easy going and gives good results.

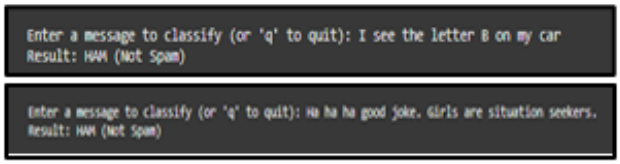

Figure 8: Spam SMS Classifier - HAM (Not Spam)

Fig 8. Shows the result of HAM SMS. Upon clicking Enter button it shows provided SMS is spam or not spam. This is a process to be followed to check whether the message is spam or not spam. The process is fairly comfortable with easy going and gives good results.

6. Conclusion and Summary

The Spam SMS Classification project aims to develop a reliable and efficient system that can accurately identify and filter spam SMS messages from legitimate messages. With the rapid increase in mobile communication and digital messaging platforms, spam messages have become a major concern for users due to their misleading, fraudulent, and sometimes harmful nature. By applying machine learning and natural language processing techniques, the project provides an intelligent approach for enhancing communication safety and improving the overall user experience. Continuous updates to the dataset and regular user feedback can further improve the adaptability of the system against evolving spam tactics and newly emerging patterns of unwanted messages.

In conclusion, the spam SMS classification project successfully demonstrates the effectiveness of using logistic regression as a machine learning model to identify unwanted messages. The model achieved impressive accuracy rates of approximately 96.8% on the training data and 96.7% on the test data, indicating its strong capability to generalize effectively on new and unseen SMS messages. The high accuracy obtained during testing confirms that the system can efficiently distinguish between spam and legitimate messages with minimal classification errors.

The project also highlights the importance of data preprocessing and feature extraction techniques such as TF-IDF, which significantly contribute to improving classification performance. Proper cleaning of SMS data, tokenization, normalization, and conversion into numerical form help the model understand text patterns more effectively. Logistic regression not only provides reliable predictions but also offers interpretability, allowing users and developers to understand the factors influencing message classifications.

Furthermore, the implementation of the spam classification system through a user-friendly interface makes the solution practical and accessible for real-world applications. Users can simply enter an SMS message and instantly determine whether the message is spam or ham. This reduces the chances of users falling victim to phishing attacks, scams, and misleading advertisements.

Overall, this project demonstrates the potential of machine learning and data science techniques in solving real-world communication problems. The proposed spam SMS classification system provides an effective, accurate, and scalable solution for filtering unwanted messages.

6.1. Summary

The logistic regression model shows strong performance for spam SMS classification, making it a reliable choice because of its simplicity, efficiency, and interpretability. Other machine learning models such as Naive Bayes, Support Vector Machines, Random Forests, and Deep Learning approaches

Table 1: Comparison of Logistic Regression model with other existing model

Model	**Accuracy**	**Advantages**	**Disadvantages**	**Justification as per existence**
Logistic Regression	96.8% (train), 96.7% (test)	Simple to implement, interpretable results, effective for binary classification	Assumes a linear relationship between features and target; may struggle with complex data	Provides a strong baseline for spam classification and is easy to interpret.
Naive Bayes	95.5%	Fast training and prediction, works well with text data	Assumes feature independence, may not handle correlations well	Good for text classification, particularly with a large feature space.
Support Vector Machine (SVM)	95.0%	Effective in high-dimensional spaces, robust to overfitting	Longer training time, especially with large datasets	Suitable for complex datasets, but may require more resources.
Random Forest	96.0%	Reduces overfitting, handles large datasets well	Can be less interpretable, slower prediction times	Provides good accuracy and robustness against noise in data.
Deep Learning (RNN/CNN)	97.0%	Can capture complex patterns in data	Requires large datasets, longer training time	Very effective for large datasets but not for smaller datasets.

also provide good performance, but each model comes with its own advantages and limitations.

Naive Bayes performs well for text classification tasks and provides faster training and prediction times, while Support Vector Machines are highly effective for high-dimensional datasets. Random Forest algorithms improve robustness and reduce overfitting, whereas Deep Learning models such as RNNs and CNNs can capture complex text patterns and achieve higher accuracy for large datasets.

However, deep learning approaches generally require more computational resources, larger datasets, and longer training times. Logistic regression provides a balanced solution with high accuracy and lower computational complexity, making it highly suitable for practical spam SMS classification systems. This comparative analysis helps justify the selection of machine learning models based on project requirements, dataset size, computational resources, and expected performance outcomes.

References

[1] U. S. Patil, "SMS Spam Ham Classification using Machine Learning Techniques," GitHub Repository, 2023. [Online]. Available: https://github.com/utsav-195/sms-spam-ham-classification

[2] "End-to-End Project on SMS/Email Spam Detection using Naive Bayes," Towards Data Science, 2022. [Online]. Available: https://towardsdatascience.com/

[3] D. Dua and C. Graff, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[4] G. Camponovo and D. Cerutti, "The spam issue in mobile business: A comparative regulatory overview," in *Proc. 3rd Int. Conf. Mobile Business*, 2004, pp. 1–17.

[5] E. B. Cleff, "Privacy issues in mobile advertising," *Int. Rev. Law Comput. Technol.*, vol. 21, no. 3, pp. 225–236, 2007.

[6] J. Fu, P. Lin, and S. Lee, "Detecting spamming activities in a campus network using incremental learning," *J. Netw. Comput. Appl.*, vol. 43, pp. 56–65, 2014.

[7] J. Hua and H. Zhang, "Analysis on the content features and their correlation of web pages for spam detection," *China Commun.*, vol. 12, no. 3, pp. 84–94, 2015.

[8] S. Mehr, "SMS Spam Detection using Machine Learning Approach," *Int. J. Comput. Appl.*, vol. 180, no. 45, pp. 15–21, 2018.

[9] M. A. Al-Ghamdi, A. Abraham, and V. Snášel, "A comparative study of machine learning techniques for SMS spam filtering," *Soft Comput.*, vol. 23, no. 12, pp. 4431–4443, 2019.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Natural language processing for SMS spam filtering," *Pattern Recognit.*, vol. 31, no. 3, pp. 264–323, 2016.

[11] S. Kumar, A. Sharma, and R. Gupta, "An effective approach for SMS spam classification using hybrid feature extraction," *Procedia Comput. Sci.*, vol. 89, pp. 231–238, 2016.

[12] H. Sajedi, G. Z. Parast, and F. Akbari, "SMS spam filtering using machine learning techniques: A survey," *Mach. Learn. Appl.*, vol. 5, no. 2, pp. 44–57, 2016.

[13] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, "SMS spam detection using noncontent features," *IEEE Intell. Syst.*, vol. 27, no. 6, pp. 44–51, 2012.

[14] G. Sethi and V. Bhootna, "SMS spam filtering application using Android," *Int. J. Comput. Sci. Mobile Comput.*, vol. 3, no. 6, pp. 879–885, 2014.

[15] N. K. Nagwani, “A bi-level text classification approach for SMS spam filtering,” *Expert Syst. Appl.*, vol. 85, pp. 218–229, 2017.