Machine Learning Approaches for SMS Spam Detection. A Comparative Analysis

Priyanshu[1], Sarthak Mittal[2], Sanchit Vasdev[3]
[1,2,3] *Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, New Delhi, India*

***Abstract*—The exponential growth of Short Message Service (SMS) has led to a significant increase in unsolicited commercial advertisements, commonly known as SMS spam, particularly prevalent in regions like Asia. Developing effective SMS spam filtering systems presents challenges due to the limited availability of real SMS spam databases and the short length and informal language of messages, which hinder traditional filtering algorithms. To address these issues, this project leverages a publicly available SMS spam dataset from the UCI machine learning repository. Through rigorous feature extraction and pre-processing techniques, including tokenization, stop word removal, lemmatization, and normalization, the data is prepared for classification. This study employs and compares the performance of various machine learning algorithms, specifically K-Nearest Neighbour (KNN), Logistic Regression (LR), and Random Forest (RF), to classify SMS messages as either spam or legitimate. Our experimental results demonstrate that the Random Forest algorithm achieved the highest accuracy of 97.7%, with a precision of 97.5%. The Logistic Regression model achieved 95.1% accuracy and 92.3% precision, while K-Nearest Neighbour showed 90.3% accuracy and 100% precision. This research contributes to advancing spam filtering techniques by addressing the unique challenges of SMS communication, paving the way for more robust and accurate spam detection systems.**

I. INTRODUCTION

SMS (Short Message Service) is a widely used mobile communication protocol that allows users to exchange messages without requiring an internet connection. It has gained popularity due to its low cost, accessibility, and efficiency compared to email services [1]. However, these advantages have also attracted criminals who exploit SMS as a tool for malicious activities, causing problems for both customers and service providers [2, 3]. SMS spam refers to unwanted or unsolicited text messages sent to users' mobile phones, typically containing various types of content This study aims to address the task of classifying mobile messages as either legitimate (Ham) or spam for users. To achieve this, messages are incorporated into the existing SMS dataset, and different machine learning classifiers are analysed on a large corpus of SMS messages from individual such as advertisements, fake awards, free services, and promotions. The primary objective of spammers is to steal sensitive user information like usernames, passwords, and credit card details. They employ various strategies, and SMS messages have become one of their straightforward tactics. Phishing attacks, a common form of online attack, traditionally occur through email. However, the simplicity and widespread use of mobile phones have led phishers to consider SMS messages as a suitable method. In phishing
attacks via SMS, phishers send malicious URLs and lure users into visiting them, intending to extract sensitive personal information from their mobile phones.

This study introduces a robust approach to classify SMS messages as spam or legitimate, aiming to address the aforementioned challenges and achieve a high detection rate. Our proposed methodology focuses on three key aspects

1. Understanding the inherent characteristics and linguistic patterns of both spam and legitimate messages within a real-world dataset
2. Applying rigorous data pre-processing and feature extraction techniques to derive highly relevant features from message content, which are then used to construct feature vectors.
3. Employing and comparatively evaluating various machine learning algorithms, specifically K-Nearest Neighbour (KNN), Logistic Regression

(LR), and Random Forest (RF), to enhance message classification accuracy and achieve optimal performance. By comprehensively analysing message characteristics, extracting pertinent features, and leveraging these established machine learning algorithms, our method offers an effective solution for the accurate detection of SMS spam messages. The subsequent sections of this paper detail the dataset used, the complete methodology including feature engineering, the implementation of the chosen algorithms, and the comprehensive analysis of their experimental results.

II. METHODOLOGY

This section begins with a description of the dataset used and the process of feature extraction. It then proceeds to explain the technical details of the proposed method.

Data Collection:
For this research, we utilized a publicly available SMS Spam Collection Dataset from the UCI machine learning repository. The dataset comprises 5,574 text messages classified as either spam or ham (legitimate). Among these messages, 747 were labelled as spam, while 4,827 were categorized as ham. Each message in the dataset consists of two parts: the message string and its corresponding category

Data Preprocessing
For data pre-processing in the research, the following steps were taken.
Removal of Unwanted Columns: Three extra columns that did not contain any data were identified and removed from the dataset to reduce noise in the model.
Cleaning of Text Messages: The text messages underwent several cleaning steps:

a Tokenization. The words in the messages were split into tokens using white spaces or punctuation. The NLTK library's word tokenize function was employed for this task.

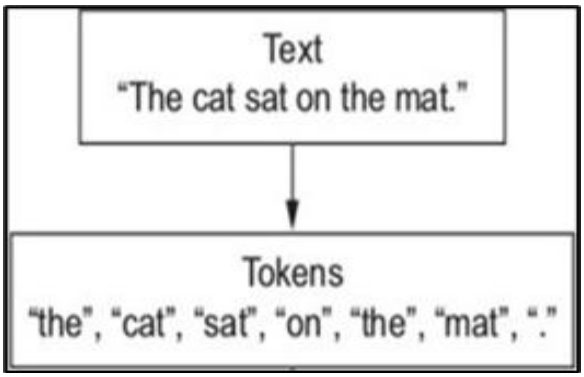

FIGURE 1: Visual Representation of Tokenization

b Removal of Stop Words. Stop words, commonly used words like "a," "an," and "the," were removed from the text messages. This step reduces dimensionality and improves model efficiency. The NLTK library's stop words package was utilized, and any stop words appearing in the predefined list were filtered out.

c Lemmatization Lemmatization involves reducing words to their base form to avoid duplication. For example, words like "study," "studying," and "studies" are considered distinct words in a Document Term Matrix, leading to increased dimensionality. The WordNetLemmatizer package from the NLTK library was used to convert inflected words to their base format.

d Normalization. Normalizing the words involves converting all words to either lowercase or uppercase to reduce word stock. For instance, "OFFER" and "offer" would be considered separate words before normalization. Python's lower () function was employed to achieve data normalization.

These pre-processing steps help to clean and transform the data in preparation for training a machine learning model on the SMS spam.

Feature Extraction and Selection
The extraction of features plays a crucial role in spam message detection, as the choice of features significantly impacts the performance of machine learning techniques. Discovering the most relevant and effective features for efficiently classifying SMS spam messages can be a challenging task. Therefore, it is essential to select features that exhibit strong

correlation to improve detection rates and reduce processing time. In the feature extraction phase, we utilized a framework to read lines from the dataset and extract features from the text messages. These features were then saved in a new structure. To identify the most efficient features for SMS spam message detection, we conducted a thorough investigation of various characteristics specific to spam messages. We selected features that were essential and helpful in identifying such messages within our dataset.

Table 1 provides an overview of the extracted features utilized in our study.

TABLE 1. Description of selected features

Index	Feature	Feature Description
1	URLs	The existence of URL in the message
2	Punctuation marks	The existence of dot and comma symbols in a message seems to be a good sign for legitimate messages since people use dots for separate sentences and chatting.
3	Mathematical Symbols	Spammers usually use mathematical symbols in their spam messages. Symbols like: +, −, <, >, / and ^.
4	Special symbols	The existence of special symbols in a message will usually signify that a message is spam since many spammers use these symbols for different reasons. Special symbols like: "$", "!", &, #, ~, and *.
5	Emoji symbols	Many normal people use emoji symbols in their messages, and it seems like a good sign for detecting legitimate messages. Symbols like: :), :*, :p, :-), :'(, etc.
6	Uppercased words	Many spammers usually use uppercased words to gain the user's attention. For example, UNLIMITED, AWARD, URGENT, WINNER, FREE, DATE, etc.
7	Phone Number	Many spammers usually send a phone number in a message, requesting users to call that given number, and eventually steal their personal information in the process.
8	Special Keywords	Many spam messages contain some suspicious keywords such as cash, ringtone, bonus, congrats, prize, voucher, etc. These keywords could reflect spam messages.
9	Message Length	Defined as the total number of characters in the message.
10	Number of Words	Defined as the total number of words in the message. Usually, spam messages contain a large number of words.

Table 2 describes how each attribute value was extracted from the dataset of spam and ham messages

TABLE 2. List of extracted features for spam and ham messages

Feature Name	Text Messages	
	"I like you peoples very much:) but am very shy pa." *(Legitimate message)*	"SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day,6days, 16+ TsandCs apply Reply HL 4 info" *(Spam message)*
URLs	No	No
Punctuation marks	Yes	Yes
Mathematical Symbols	No	Yes
Special symbols	No	Yes
Emoji symbols	Yes	No
Uppercased words	No	Yes
Phone Number	No	No
Special Keywords	No	Yes
Message Length	50	137
Number of Words	11	26

The initial stage of our approach involved collecting the dataset and determining the features to be used in the experiment. Subsequently, we extracted features from all SMS messages in the dataset and transformed them into feature vectors, which were then utilized to train and test our proposed model. Finally, we applied these extracted features to the proposed model to classify SMS messages. The architectural representation of our model is depicted in Figure 1.

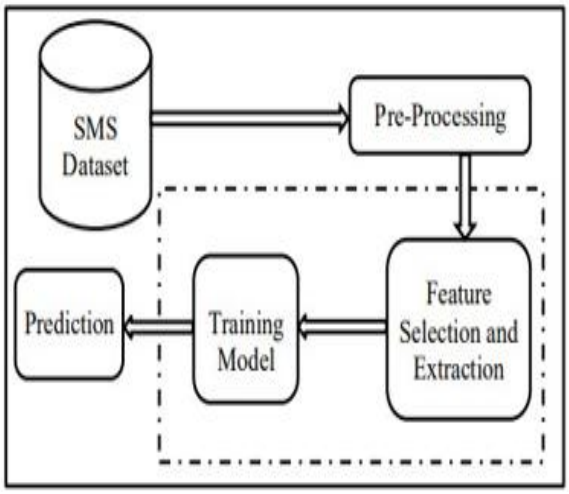

Figure 2. The structure of proposed method

III. ALGORITHMS

Machine learning tasks are broadly categorized based on their learning method and feedback. Two extensively used approaches include supervised learning, where algorithms are trained using labelled input and output data, and unsupervised learning, where patterns are discovered without labelled information. For this project, we primarily utilized supervised learning algorithms, focusing on the following approaches

1. K-Nearest Neighbour: In the K-Nearest Neighbour (KNN) algorithm, when evaluating each test data point, we consider its K nearest training data points. By examining these nearest neighbours, the most frequently occurring class among them is determined and assigned to the test data point. The value of K thus represents the number of training data points used to determine the category of a new instance.
2. Logistic Regression: The logistic function is employed in this particular machine learning algorithm, where the dependent variable is categorical. It quantifies the relationship between the independent variable and the categorical dependent variable.
3. Random Forest: Random Forest is an extensively utilized supervised learning algorithm in machine learning. It is applicable for both Classification and Regression tasks. This algorithm operates on the principle of ensemble learning, which involves the combination of multiple classifiers to address complex problems and enhance the model's performance.

IV. EVALUATION MATRIX

There are several metrics which are considered to evaluate the effectiveness of our proposed approach: true positive rate, false positive rate, true negative rate, false negative rate, F1 score, accuracy, precision, and recall. These metrics are commonly used to assess the performance of spam detection systems. Here is a brief explanation of each metric.

a True Positives (TP): It represents the number of spam messages correctly classified by the machine learning algorithm.

b True Negatives (TN): It indicates the number of non-spam (ham) messages accurately classified as ham by the machine learning algorithm

c False Positives (FP): It denotes the number of ham messages wrongly classified as spam by the machine learning algorithm (Type 1 Error)

d False Negatives (FN): It represents the number of spam messages incorrectly classified as ham by the machine learning algorithm (Type 2 Error)

e Accuracy: This is the proportion of total correct predictions (both true positives and true negatives) among the total number of cases examined. It can be calculated as

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Population}$$

f Precision: It indicates the percentage of messages classified as spam that are actually spam. Precision measures the exact correctness and can be calculated as

$$Precision = \frac{True\ Positives}{Predicted\ Positive\ (TP + FP)}$$

g Recall (Sensitivity): It represents the percentage of actual spam messages that are correctly classified as spam. Recall measures the completeness and can be calculated as:

$$Recall = \frac{True\ Positives}{Actual\ Positive(TP + FN)}$$

h F1-Score: It is defined as the harmonic mean of Precision and Recall, providing a balanced measure of the algorithm's performance. It can be calculated as.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision \pm Recall}$$

i Receiver Operating Characteristics (ROC) Area: It refers to the area under the curve plotted between True Positive Rate and False Positive Rate for different threshold values

These metrics collectively provide valuable insights into the performance and accuracy of the spam detection system

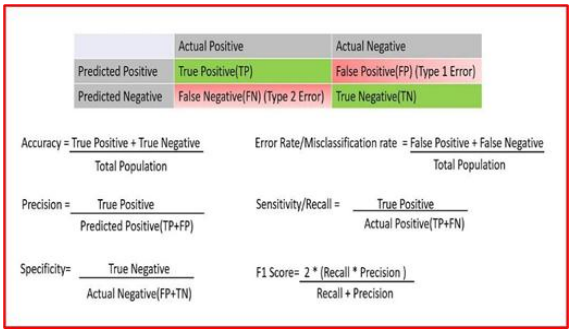

Figure 3: Confusion Matrix and Evaluation Metrics

V. RESULTS AND DISCUSSION

The performance of the three machine learning models K-Nearest Neighbour (KNN), Logistic Regression (LR), and Random Forest (RF) was evaluated on an unseen test set to assess their generalization capabilities. The complete results, including accuracy, precision, recall, and F1-score, are presented in Table 3.

Algorithm	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbour (KNN)	90.40%	100%	28.20%	44.00%
Logistic Regression (LR)	95.10%	92.00%	69.10%	78.90%
Random Forest (RF)	97.70%	97.70%	84.60%	90.70%

Table 3: Performance Comparison of Classification Models on the Test Set

The Random Forest (RF) classifier emerged as the most effective and balanced model, achieving the highest accuracy of 97.7% and the highest F1-Score of 90.7%. Its strong precision of 97.7% indicates that when it identifies a message as spam, it is almost always correct. Furthermore, its recall of 84.6% is the highest of the three models, meaning it successfully identified the vast majority of all spam messages in the test set. This superior performance is attributable to Random Forest's ensemble nature, which combines predictions from multiple decision trees to reduce overfitting and improve robustness. The Logistic Regression (LR) model also demonstrated strong performance, with an accuracy of 95.1% and an F1-Score of 78.9%. While its precision was a respectable 92.0%, its recall of 69.1% shows that it failed to identify nearly a third of the spam messages, making it less reliable for comprehensive spam filtering compared to Random Forest.

Interestingly, the K-Nearest Neighbour (KNN) model achieved a perfect precision of 100%, meaning every single message it flagged as spam was indeed spam (zero false positives). However, this came at a significant cost to its recall, which was only 28.2%. This indicates that while KNN is extremely cautious and accurate when it flags a message, it misses over 70% of the actual spam, leading to a low F1-Score of 44.0% and the lowest overall accuracy of 90.4%. This makes it unsuitable for practical use were catching as much spam as possible is a key objective.

In conclusion, the empirical evidence clearly demonstrates that Random Forest provides the best trade-off between identifying spam (recall) and being correct when doing so (precision). Its high F1-Score confirms it as the most robust and reliable algorithm for this SMS spam detection task.

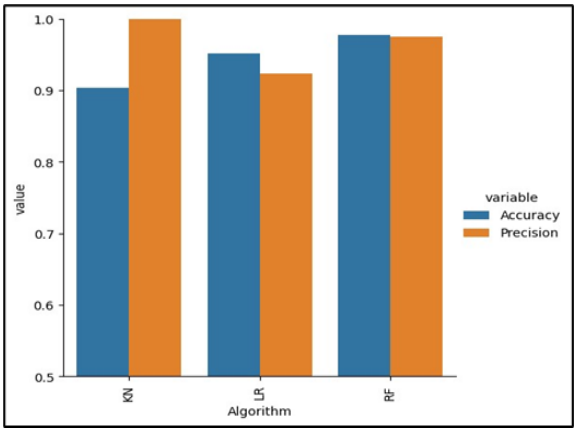

Figure 4: Accuracy & Precision

VI. CONCLUSION

This study successfully developed and evaluated a machine learning system for SMS spam detection, demonstrating the effectiveness of supervised learning in accurately classifying mobile messages. Through a robust methodology involving data pre-processing, feature extraction, and comparative analysis, this research addressed the key challenges of SMS spam filtering.

The experimental results on an unseen test set conclusively identified the Random Forest (RF) classifier as the most effective model, achieving a superior accuracy of 97.7% and a balanced F1-Score of 90.7%. While the K-Nearest Neighbour (KNN) model yielded perfect precision, it's extremely low recall rendered it impractical. The Random Forest model, by contrast, provided the best compromise between high precision (97.7%) and high recall (84.6%), ensuring that most spam is caught with very few errors. These findings highlight the suitability of ensemble methods like Random Forest for creating reliable and effective spam detection systems. The successful implementation of this system contributes to a safer user experience by mitigating the security risks associated with SMS spam.

VII. FUTURE SCOPE

The future scope of spam alert detection systems holds significant potential for advancements and

improvements. Here are several key areas that offer promising opportunities for further development

Deep Learning and Neural Networks: The application of deep learning techniques, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), holds promise for enhancing spam detection. Neural networks can automatically learn complex patterns and features from raw data, allowing for more accurate and adaptive spam detection models. Natural Language Processing (NLP) and Contextual Analysis: Leveraging advanced NLP techniques, such as semantic analysis, sentiment analysis, and contextual understanding, can enhance the ability to detect spam messages that may employ sophisticated language patterns or deceptive tactics. Incorporating contextual analysis can improve the accuracy of spam detection by considering the message's context and intent.

Cross-Platform Spam Detection: With the increasing use of various communication channels, including emails, text messages, social media platforms, and instant messaging apps, the future scope lies in developing spam detection systems that can effectively operate across multiple platforms. This holistic approach would provide comprehensive protection against spam across different communication channels.

Real-Time Detection: The ability to detect and prevent spam in real-time is crucial in addressing emerging spamming techniques and minimizing their impact. Future spam alert detection systems should focus on reducing detection latency and providing immediate alerts or actions to users, ensuring timely protection against spam messages.

Privacy-Preserving Techniques: As privacy concerns become more prominent, future spam detection systems should explore privacy-preserving techniques that maintain user privacy while still effectively detecting spam. Approaches like federated learning, secure multi-party computation, and differential privacy can be explored to strike a balance between privacy and spam detection effectiveness.

Adaptive and Self-Learning Systems: Developing spam alert detection systems that can adapt and learn from evolving spamming techniques is essential. These systems should be capable of continuously updating their spam detection models based on new patterns and techniques employed by spammers, ensuring ongoing effectiveness in spam prevention.

User-Centric Customization: Allowing users to customize spam detection preferences and thresholds can enhance their control over the filtering process. Future systems can incorporate user feedback mechanisms to improve personalization and tailor spam detection based on individual preferences and priorities.

REFERENCES

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

[2] E. B. Cleff, "Privacy issues in mobile advertisin'," *Int. Rev. Law Comput. Technol.*, vol. 21, pp. 225-236, 2007

[3] 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, 2015

[4] J. Hua and Z. Huaxiang, "Analysis on the content features and their correlation of Web pages for spam detection," *China Commun. *, vol. 12, no. 3, pp. 84-94, 2015

[5] B. Reaves, N. Scaife, D. Tian, L. Blue, P. Traynor, and K. R. Butler, "Sending out an SMS: Characterizing the security of the SMS ecosystem with public gateways," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2014, pp. 339-356.

[6] C. Wang et al., "A behavior-based SMS antispam system," *IBM J. Res. Develop. *, vol. 54, no. 6, pp. 3:1-3:16, 2010.

[7] T. Yamakami, "Impact from mobile SPAM mail on mobile internet services," in *Parallel and Distributed Processing and Applications*. Berlin, Germany: Springer, 2004, pp. 179-184

[8] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27-39, 2019.

[9] S. W. Liew, N. F. M. Sani, M. T. Abdullah, R. Yaakob, and M. Y. Sharum, "An effective security alert mechanism for real-time phishing tweet detection on Twitter," *Computers and Security*, vol. 83, pp. 201-207, 2019. Doi: 10.1016/j.cose.2019.02.004.

[10] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," *Int. J. Comput. Sci. Netw. *, vol. 2, 2013.

[11] J. Ma, Y. Zhang, J. Liu, K. Yu, and X. Wang, "Intelligent SMS spam filtering using topic model," in *2016 International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, 2016, pp. 380-383.

[12] E.-A. Minastireanu and G. Mesnita, "Light GBM Machine Learning Algorithm to Online Click Fraud Detection Light GBM Machine Learning Algorithm to Online Click Fraud Detection," *Journal of Information Assurance & Cybersecurity*, vol. 3, pp. 1-15, Apr. 2019.

[13] H. Najadat, N. Abdulla, R. Abooraig, and S. Nawasrah, "Mobile SMS Spam Filtering based on Mixing Classifiers," *International Journal of Advanced Computing Research*, vol. 1, 2014.

[14] T.-h. Pham, "Content-based Approach for Vietnamese Spam SMS Filtering," in *2016 International Conference on Asian Language Processing (IALP)*, 2016, pp. 41-44.

[15] A. Prieto, B. Prieto, E. Ortigosa, E. Ros, F. Pelayo, J. Ortega, and I. Rojas, "Neural networks: An overview of early research, current frameworks and new challenges," *Neurocomputing*, vol. 214, 2016.

[16] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," *Information Sciences*, vol. 385-386, pp. 213-224, 2017.

[17] P. Sethi, V. Bhandari, and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," in *2017 International Conference on Computing and Communication Technologies for Smart Nation, IC3TSN 2017*, Oct. 2017, pp. 28-31.