SMS Secure: Machine Learning as the Guardian Angel Against Spam

Arwaa Mamdoh, Abdelrahman Mohamed Aly,
Mayar Adel, Nader Maged, Yassmin Ezzat Aly
Faculty of Computer Science
Misr International University, Cairo, Egypt
arwaa2110478, abdelrahman2110423,
mayar2106643, nader2110520, yassmin2104022{@miuegypt.edu.eg}

***Abstract*—Recognizing the pervasive issue of SMS spam, there is a critical need for robust detection mechanisms. Despite continual advancements in the field, achieving accurate SMS spam detection remains a challenging endeavor. In response to this challenge, our paper introduces an adept framework designed for efficient SMS spam detection. We leverage the capabilities of six machine learning algorithms or methods, namely SVM, kNN, Naïve Bayes, Random Forest, Logistic Regression,and Decision Tree. This framework is rigorously evaluated across three diverse datasets: "Spam SMS classification," "SMS spam collection Dataset," and "Spam/Ham SMS data."To assess the performance of our framework, we employ standardized metrics such as accuracy, recall, precision, and F1 score. Notably, the SVM, Support Vector Machine, consistently emerges as the top-performing algorithm. Specifically, in the "Spam SMS classification" dataset, SVM achieves an exemplary F1 score of 0.69. Transitioning to the "SMS spam collection Dataset," SVM leads with the highest accuracy, registering an impressive 0.90. The trend continues in the "Spam/Ham SMS DataSet," where SVM once again excels with a commendable accuracy of 0.90.This success is attributed to the strategic utilization of SVM, Although it gives the same percentages as Random Forest and Decision Tree algorithms in the 3 datasets but capitalizes on its proficiency in handling high-dimensional spaces, managing non-linear relationships, and exhibiting a faster runtime when compared to Random Forest and Decision Tree algorithms. The efficacy of our framework across diverse datasets underscores the significance of thoughtful algorithm selection in crafting an adept SMS spam detection framework.**

Keywords: Machine Learning; SMS Spam Detection; Classification; Support Vector Machine kNN; Random Forest; Naïve Bayes; (SVM); Logistic Regression; Decision Tree; Spam/Ham Classification; SMS Spam Collection.

I. INTRODUCTION

The ubiquity of Short Message Service (SMS) communication in our daily lives has made it a prime target for spam activities, necessitating effective detection mechanisms. As mobile devices continue to play a central role in interpersonal communication, the infiltration of the spammed messages not only disrupts user experience but also poses security and privacy threats. In light of this, the significance of developing robust SMS spam detection systems cannot be overstated.

SMS spam remains a persistent challenge despite advancements in communication technologies. The problem is further exacerbated by the increasing sophistication of spam tactics. According to recent studies , spam messages constitute a significant percentage of all SMS traffic, causing inconvenience to users and potentially leading to security breaches. The need for accurate and efficient SMS spam detection is underscored by the growing volume and potential impact of these unwanted messages.

Machine learning, with its ability to learn patterns and make predictions from data, emerges as a promising avenue for addressing the SMS spam detection problem. Leveraging machine learning algorithms allows us to analyze the intricate patterns and linguistic nuances in SMS messages, enhancing the accuracy and adaptability of detection systems. As we delve into this research, the application of machine learning in tackling SMS spam promises to revolutionize the way we mitigate this pervasive issue.

The main contribution of this paper can be summarized as follows: We present an efficient framework employing machine learning techniques for the detection of SMS spam. Our approach is rigorously evaluated across three diverse datasets: "Spam SMS classification," "SMS spam collection," and "Spam/Ham SMS" datasets. Furthermore, the results reveal the superior performance of the Support Vector Machine (SVM) algorithm, with SVM consistently outperforming other algorithms in terms of precision, accuracy, recall, and F1 score across all datasets. This paper not only provides insights into the effectiveness of machine learning in SMS spam detection but also establishes a benchmark for the selection of the most suitable algorithm for different datasets.

The paper explores earlier research in the "Related Work" section, providing insights into the field. The "Methodology" section details dataset acquisition and preparation, establishing a solid research foundation. In the "Results" section, algorithm outcomes are presented, confirming the algorithm with the highest accuracy. The "Conclusion" section succinctly summarizes key findings and potential implications. An

acknowledgment of all the supporting figures of this research is presented in the sixth section.

II. Related Work

The SMS spam Detection Model is investigated by many people who came up with numerous satisfactory results. We will mention the papers we read and assisted us in our research. Moreover, all the papers we mentioned down below will be referenced at the end in the references section.

In [1], the authors addressed the issue of SMS spam and conducts a comparative study of machine learning classifiers for its detection. The main objective is to assess the performance of various algorithms and investigate the impact of bag-of-words (BOW) and TF-IDF methods. The study involves dataset pre-processing, feature extraction using BOW and TF-IDF, and evaluation of classifiers such as MLP, SVM, random forest, and KNN based on accuracy, F-measure, precision, recall, and ROC curve. Results indicate that MLP outperforms other algorithms, particularly with the BOW method. However, the study is limited by its use of a single dataset, exclusive focus on BOW and TF-IDF, and lack of exploration into different parameters' impact on classifier performance.

In [2], the authors aimed to address spam in mobile message communication by proposing a machine learning-based approach for accurate and efficient detection. The method uses classifiers like K-nearest neighbor, decision trees, and Logistic regression, to classify ham and spam messages. The results show a remarkable 99% accuracy, with the Logistic regression classifier showing particularly noteworthy performance compared to K-nearest neighbor and decision trees. The study also discusses the potential implications of implementing machine learning-based spam detection methods for secure mobile communication. However, there are certain drawbacks, such as data accessibility and possible difficulties with practical applications.

Machine Learning Sms Spam Detection Model[3] The research paper focuses on the issue of SMS spam in Kenya's mobile commerce and emphasizes the importance of having reliable and cost-effective technology to detect and prevent it. The primary aim of the research is to assess a machine-learning SMS spam detection model, which involves creating a spam detection model, demonstrating the use of machine learning for message classification, and testing the model through a prototype. The Naive Bayes method is utilized to detect spam SMS, and various measurement methods are used to compare classification results. The study reveals that the Naive Bayes method achieves a high accuracy of 96.1039% in the classification of the instances, indicating the effectiveness of the machine learning model in detecting SMS spam. However, the research may have limitations due to its focus on a specific machine learning algorithm and the prototype's scope, suggesting the need for further exploration of alternative approaches to SMS spam detection.

In [4]. The paper discusses the problem of identifying spam messages in SMS conversations using a modified Transformer model The study evaluates the model's performance against conventional machine learning classifiers and LSTM deep learning on Twitter and datasets of SMS spam collection. The proposed model demonstrates improved accuracy, recall, and F1-Score, particularly in handling both balanced and imbalanced datasets. However, the paper acknowledges challenges, such as the impact of unfamiliar words in SMS texts on model predictions.

The paper[5] "SMS Spam Filtering Application Using Android" investigates the application of Bayesian filtering techniques to combat the issue of SMS spam. It delves into various methods, such as the Bayesian algorithm and content-based filtering, outlining stages that involve training and classification processes. The research outcomes highlight the efficacy of utilizing known messages for training purposes and assessing the spam filter's performance through measures like sensitivity and specificity. However, the paper falls short in explicitly addressing limitations or challenges encountered during the implementation of these techniques. This gap opens avenues for further exploration to uncover potential drawbacks or areas that may benefit from enhancements in the proposed approach.

The research addresses the prevalent issue of SMS spam. In [6], striving to create a robust method for detecting spam and protecting individuals from potential financial and privacy risks. The proposed approach utilizes Natural Language Processing (NLP) and Deep Learning techniques, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The objectives include preparing data, constructing the model, evaluating performance using metrics like accuracy and spam capture rate, and comparing it with machine learning models. The process involves collecting data from an SMS spam dataset, preparing it using NLP, and developing models with LSTM and GRU. Although the results demonstrate the superior performance of LSTM and GRU models, the study acknowledges limitations, such as its focus on English.

This paper explores advanced techniquesIn [7] like the transformer model for the detection of SMS spam and recurrent neural networks (RNNs). It introduces a customized transformer model for the identification of SMS spam, showcasing superior performance compared to traditional classifiers and other deep-learning models. While acknowledging the limitations of traditional methods, the research highlights the robustness of deep learning. Despite challenges with recurrent neural networks (RNNs), the proposed model stands out as a promising approach for future

studies.

The Research paper[8] discusses the pervasive issue of spam emails on the internet and the challenges posed by their high volume. The majority of emails received by users are classified as spam, making manual analysis an impractical solution. To address this issue, researchers are developing a classifier that uses machine learning methods to accurately detect and reduce spam emails,as they aim to achieve high accuracy in classification of spam through the use of several methods including data analysis and classification algorithms such as Support Vector Machines, Naive Bayes, Decision Trees, and to classify the emails as spam or non-spam,Through the use of these methods, the researchers hope to develop a highly accurate spam classifier that can reduce the number of spam emails received by users. This will help ensure that users receive only relevant and important emails in their inbox, improving their productivity and overall email experience.

In [9], the authors addressed the problem of spam SMS messages and the need for effective detection methods to protect private data. The objective is to compare different machine learning classifiers on two datasets and evaluate their performance based on accuracy, recall, CAP curve, and precision. The methods used include traditional machine learning techniques and deep learning methods like Convolutional Neural Networks. The results show that SVM and Naive Bayes classifiers perform well on the datasets, while CNNs also show promise. However, the presence of informal language and short length of text messages are limitations that make spam SMS detection challenging.

In [10], the authors explored The paper managed to explore into the problem of SMS spam detection and aims to explore the effectiveness of CNN in categorizing SMS spam messages. The study utilized Tiago's dataset and implemented a CNN model using python programming language in the Spyder IDE. The evaluation of the model was conducted using a 10-fold cross validation and performance measures such as AUC, confusion matrix, and F1 score. The results indicate that the CNN model achieved high sensitivity and specificity values, demonstrating its effectiveness in SMS spam detection. However, the study has some limitations, the use of only one dataset and the lack of comparison with other state of the art models.

In [11], the authors managed to address the problem of SMS spam has become a significant security threat in today's mobile phone era. The objective of this study was to detect and prevent sms spam using numerous machine learning techniques. The methods used in the study included K Nearest Neighbor, Support Vector Machine, Random Forest, Naïve Bayes,and Logistic Regression. The results of the study showed that SVM achieved the highest accuracy rate reaching 99%. However, the study's limitations include the scarcity of reliable datasets and the difficulty in developing algorithms due to the expansion of idioms and abbreviations in text messages.

In [12], the authors addressed the challenge of spam email detection and developed an efficient classification model to defferntiate between spam emails and legitimate emails. Various machine learning classifiers such as SVM, Bayesian with Adaboost,Bayesian, Naïve Bayes, J48, and Naïve Bayes with Adaboost are compared and evaluated based on metrics including F-measure, False Positive Rate, and training time. The results reveal that SVM is the most accurate classifier due to its low false positive rate and high accuracy, despite the longer training time. However, the study is limited by the removal of legitimate emails by the filters, and the challenges of distinguishing between spam and legitimate emails based on subject or content.

In [13], the authors addressed the problem of SMS spam by employing supervised machine learning algorithms, including support vector machines, naïve Bayes, and maximum entropy, to classify messages as spam or ham. The goal is to compare the performances of these algorithms in accurately filtering spam and ham messages. The methods involve building models to classify messages and analyzing the accuracy of each algorithm. The results show that the support vector machine algorithm provides the most accurate outcomes, with an accuracy of up to 98%, followed by naïve Bayes and maximum entropy algorithms. However, the study's limitations include potential challenges in handling short and slang-filled SMS texts, which may affect the effectiveness of the classification models.

The paper [14] is about developing a model to accurately identify SMS spam messages based on their characteristics. It reviews existing methods and proposes a new methodology that involves data collection, feature extraction and selection, and an averaged neural network. The paper presents experimental results that show the proposed model outperforms some other methods in terms of accuracy and F-measure, and can effectively identify SMS spam messages. The conclusion suggests possible improvements and future directions for the research.

The paper [15] proposes an intention-based approach for detecting SMS spam messages using NLP models like BERT, DistilBERT, and SpanBERT as they aim to provide an accurate and efficient classification of SMS messages as spam or non-spam. The paper highlights the effectiveness of the different models and their combinations in accurately detecting, with DistilBERT showing particularly promising results. However, the study has some limitations, including a limited dataset, a limited set of features, and limited language support. The authors suggest that future studies should consider larger and more diverse datasets, explore additional features, and evaluate the proposed approach on

other languages.

The paper [16] compares two methods of spam filtering for Nepali text messages: Naïve Bayes and Support Vector Machine (SVM). The paper proposes a framework for spam filtering that consists of three steps: preprocessing, feature extraction, and classification. The paper uses a term frequency-inverse document frequency (TF-IDF) scheme to represent the messages as feature vectors in a vector space model. The paper evaluates the performance of both methods on a manually created Nepali SMS dataset using metrics such as accuracy, precision, and recall. The paper concludes that Naïve Bayes is a better classification technique than SVM-based classifier, achieving an accuracy of 92.74 compared to 87.15 for SVM .

In this paper [17]The authors proposed an efficient framework for detecting SMS spam messages, employing the multinomial naive Bayes with Laplace smoothing and SVM with a linear kernel. The dataset from UCI Machine Learning was utilized, showcasing a reduction of more than half in the overall error rate compared to the original paper. This improvement is attributed to factors such as meaningful feature addition, learning curve analysis, and addressing misclassified data. The results demonstrate the effectiveness of the proposed method in enhancing accuracy and mitigating challenges in SMS spam detection.

In this paper [18] The authors proposed an efficient framework for detecting SMS spam messages, employing various machine learning algorithms such as NB, DT, RF, and SV. They utilized a collected dataset of SMS messages and evaluated algorithm performance using accuracy, precision, recall, and F1 score. Results indicated that Random Forest outperformed others with an accuracy percentage of 98.5. The study acknowledges limitations, including the need for a large dataset and the risk of false positives, offering valuable insights for future research in SMS spam detection using machine learning.

In this paper [19] The paper proposes an H2O-based framework for SMS spam detection, comparing machine learning algorithms. Evaluating RF, Deep Learning, and NB on a UCI dataset, RF outperforms with 97.7 accuracy percentage, 96 precision percentage, and 86 recall percentage. Despite a longer runtime (30.28 seconds), RF's overall performance stands out. The study emphasizes feature extraction's role in improving classifier accuracy and discusses the importance of balanced metrics for cybersecurity. Overall, the paper provides valuable insights for enhancing SMS spam detection systems and contributes significantly to information security.

in this paper [20] The study addresses the increasing issue of SMS spam messages due to the growing popularity of mobile phones. The objective is to develop an effective spam filtering technique using machine learning algorithms. The methods involve data collection, preprocessing, feature extraction, and model selection, with a focus on the Naive Bayes algorithm. The results indicate that the TF-IDF with Naive Bayes outperforms other algorithms in terms of accuracy. However, the study acknowledges the limitation of imbalanced datasets affecting the evaluation of performance.

III. Proposed Methodology

Numerous algorithms were used, and a research was done on each algorithm before training the model using them on the datasets.

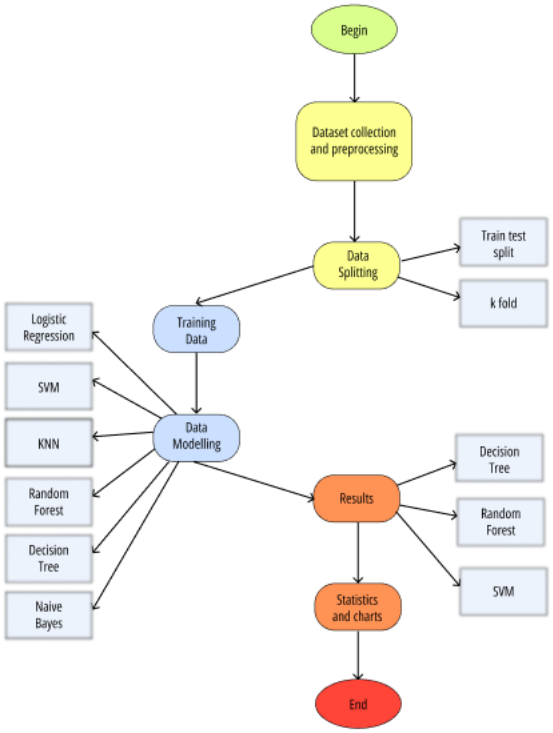

Fig. 1. Flow Chart

A. Datasets Descriptions

The first dataset consists of 2 features[21], and it has 5,169 records. The dataset was split into two partitions: 75% for training, and 25% for testing,also K-fold with 5. A detailed description of the features can be found below.

The dataset is splitted to messages and its type, whether it is Spam or not, Spam which are unwanted and ham which are important to read and we know by many processes before using models.

TABLE I
FEATURES OF DATASET 1

Feature	Type	Values
Type	Classification	Ham or Spam
SMS	Characters	Any characters

The second dataset consists of 2 features[22], and it has 5,156 records.The dataset was normalized, then it was split into two partitions: 75% for training, and 25% for testing,also K-fold with 5. A detailed description of the features can be found below.

Spam or Ham is also the target, and it represents whether the message is important or not. Many processes are taken to know this either by calculating the length of the messages or by specific keywords that are used in each case.

TABLE II
FEATURES OF DATASET 2

Feature	Type	Values
Type	Classification	Ham or Spam
SMS	Characters	Any characters

The third and final dataset consists of 2 features[23], and it has 4193 records.The dataset was normalized, then it was split into two partitions 75% for training, and 25% for testing,also K-fold with 5.It is the same as the previous 2 Datasets.

TABLE III
FEATURES OF DATASET 3

Feature	Type	Values
Type	Classification	Ham or Spam
SMS	Characters	Any characters

B. *Pre-processing*

The code defines a function called clean-text that takes a string as input which is the message and returns a cleaned version of the string. The function first converts the string to lowercase, removes all non-alphanumeric characters, removes any extra spaces, and finally removes all stopwords from the string. Stopwords are common words that do not carry much meaning, such as “the”, “and”, and “is”. The set of stopwords used in the code is defined as a set of common English stopwords. The cleaned string is then returned by the function. This pre-processing step can be useful for text analysis tasks such as sentiment analysis, topic modeling, and text classification.

C. *Used Algorithms*

The datasets underwent analysis using six distinct machine learning techniques: SVM, K Nearest Neighbor (k-NN), Logistic Regression, Random Forest, Naive Bayes, and Decision Tree. For each method, metrics such as Accuracy, Precision, Recall, and F1-Score were computed. The subsequent sections of the paper contain the compiled results, visual representations in charts, and an in-depth examination and comparison of these outcomes.

1) **SVM:**
The Support Vector Machine [24] is a versatile machine learning algorithm applicable to both regression and classification tasks, although it’s particularly well-suited for classification purposes, as in our case. Its primary aim is to discover the optimal hyperplane within an N-dimensional space, effectively separating points into distinct classes within the feature space. This hyperplane is specifically designed to maximize the margin between the closest points of different classes. The dimensionality of this hyperplane corresponds to the number of features: in a scenario with two input features, the hyperplane manifests as a line, while with three input features, it becomes a 2-D plane. $\frac{w^T(x_{\text{pos}}-x_{\text{neg}})}{\|w\|} = \frac{2}{\|w\|}$

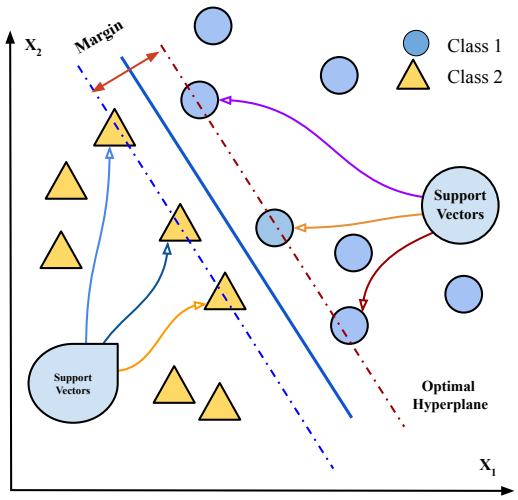

Fig. 2. Illustration of SVM [25]

2) **Decision Tree:**
Supervised learning methods encompass the Support Vector Machine [26] and decision tree algorithms, which cater to both regression and classification tasks. In decision trees, each node represents a class label, while the inner nodes showcase attributes. These trees effectively capture Boolean functions with discrete features. When employed in a decision tree, a node’s utilization alters the entropy, effectively segmenting the training dataset into smaller groups. This shift in entropy signifies information gain.
To clarify: Suppose S is a set of the instances, while A

is an attribute, S_v is the subset of S with $A = v$, and Values(A) is the set of all the possible values of A, then

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \mid \frac{|S|}{S \mid} \cdot \text{Entropy}\,(S_v) \tag{1}$$

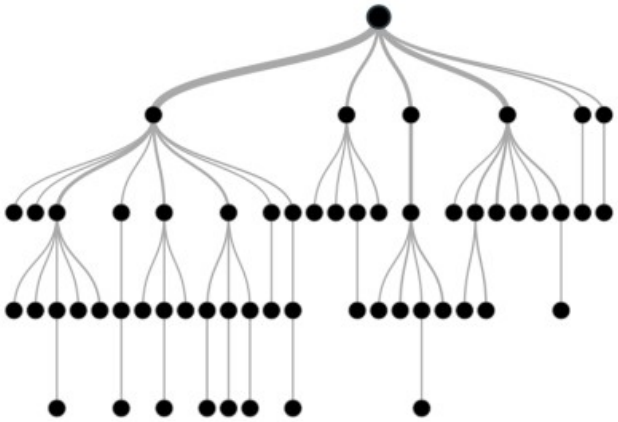

Fig. 3. Decision Tree

3) **Naïve Bayes:**
Derived from the Bayes Theorem, Naive Bayes [27] stands as a straightforward yet powerful classification algorithm. Its foundational assumption lies in predictor independence, suggesting that traits or features are unrelated or unconnected. Despite potential dependencies, each attribute independently contributes to the probability, thus earning its label "Naive."

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \tag{2}$$

$$P(c \mid X) = P\,(x_1 \mid c) \times P\,(x_2 \mid c) \times \cdots \times P\,(x_n \mid c) \times P(c) \tag{3}$$

4) **K – Nearest Neighbor:**
In 1951, Hodges and Fix introduced the K-Nearest Neighbor (KNN) rule [28], a non-parametric pattern classification algorithm. KNN stands out as a fundamental and robust classification method, particularly applicable in scenarios where minimal or no prior understanding of data distribution exists. This algorithm operates without assuming any specific data patterns and functions by associating the value of a given data point with the most near data points in the training set and assigning a target value accordingly.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{4}$$

5) **Random Forest:**

The Random Forest, [29], stands as a supervised machine learning algorithm suitable for regression and classification tasks, though it excels particularly in classification. This method involves the consideration of many decision trees before generating the results or outputs. For classification, it employs a voting mechanism to determine the class, while in regression, it relies on the mean of all decision tree outputs. Renowned for its efficiency with sizable, high-dimensional datasets, the algorithm operates on the premise that an increased number of trees leads to improved decision-making.

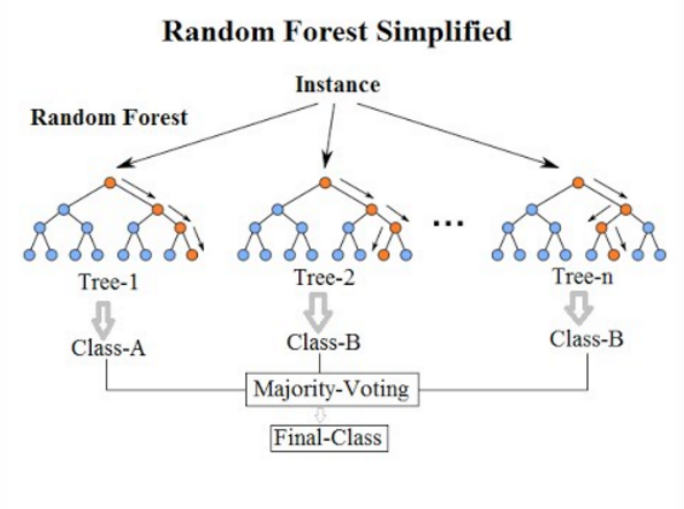

Fig. 4. Random forest demonstration

6) **Logistic Regression:**
Logistic Regression, a supervised machine learning algorithm [30], is primarily a classification technique. It is employed to analyze how feature variables impact categorical outcomes, typically involving binary labels like the presence or absence of a spam. In scenarios with multiple features, a multiple or multivariable logistic regression model stands as one of the frequently utilized predictive methods.

$$\text{Logit}(pi) = 1/(1 + \exp(-pi)) \tag{5}$$

D. *Performance Metrics*

Accuracy is the count of legitimately anticipated data from all the data. The count of accurately anticipated positives taken from the anticipated positives is the Precision Recall is the number of correctly anticipated positives from all the true positives. The number of accurately anticipated negatives out of all the expected negatives is known as specificity.

$$\text{Accuracy} = (\text{TN}+\text{TP})/(\text{TN}+\text{TP}+\text{FN}+\text{FP}) \quad (6)$$

$$\text{Precision} = \text{TP}/\text{TP}+\text{FP} \quad (7)$$

$$\text{Recall} = \text{TP}/\text{TP}+\text{FN} \quad (8)$$

$$\text{F1-Score} = 2\cdot\frac{\text{precision}\ \cdot\ \text{recall}}{\text{precision}\ +\ \text{recall}} = \frac{\text{TP}}{\text{TP}+\frac{1}{2}(\text{FP}+\text{FN})} \quad (9)$$

IV. Results and Analysis

The results collected from SVM, Logistic Regression, Naïve Bayes, k-nearest Neighbor, Random Forest, and Decision Tree are shown below.

The following results are from the first dataset.

TABLE IV
STATISTICS OF ALGORITHMS WITH 75/25 DATA SPLIT

Model	Accuracy	Precision	Recall	f1-score	runtime
Logistic Regression	0.87	0.93	0.50	0.46	1.672
SVM	0.89	0.95	0.61	0.65	2.061
k-NN	0.87	0.93	0.51	0.49	0.748
Random Forest	0.89	0.95	0.61	0.65	2.857
Decision Tree	**0.89**	**0.95**	**0.61**	**0.65**	**1.071**
Naïve Bayes	0.87	0.93	0.51	0.47	0.696

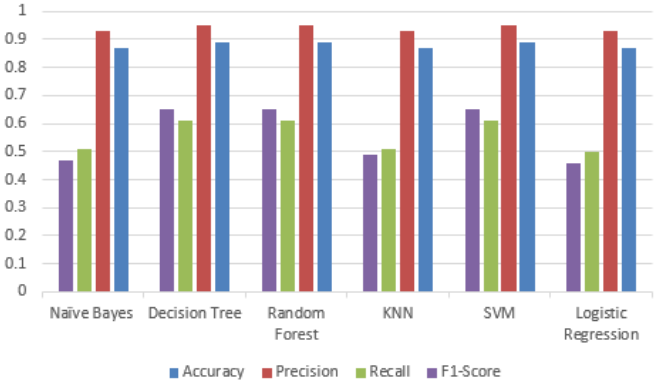

Fig. 5. First dataset performance chart with data split

Decision tree, Random forest, and SVM are the dominant algorithms, sharing the same accuracy of 0.89 with equal numbers in terms of precision and f1-score. K-NN, Naive Bayes, and LR comes in second place with slightly better numbers in term of f1-score with an accuracy of 0.87, Random forest took an extremely long time to create the model, while Naive Bayes and Decision tree were the fastest while creating the model.

TABLE V
STATISTICS OF ALGORITHMS WITH 5 K-FOLD

Model	Accuracy	Precision	Recall	f1-score	runtime
Logistic Regression	0.87	0.93	0.50	0.46	1.767
SVM	0.90	0.95	0.63	0.69	10.909
k-NN	0.87	0.93	0.54	0.49	1.467
Random Forest	0.90	0.95	0.63	0.69	22.628
Decision Tree	**0.90**	**0.95**	**0.63**	**0.65**	**7.241**
Naïve Bayes	0.87	0.93	0.50	0.47	0.643

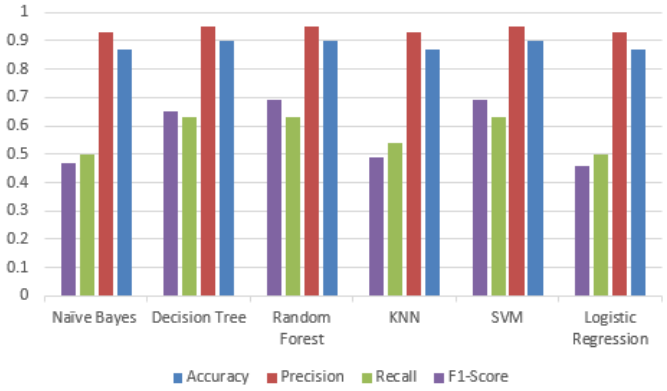

Fig. 6. First dataset performance chart with 5 k-fold

Using k-fold on this dataset had a slight level of improvement with most algorithms. The same top 3 had 0.01 of increased accuracy and 0.4 of increased f1-score. KNN had a very slight increase of 0.03 in terms of Recall. Naive Bayes suffered loss in terms of Recall.LR did not change compared to the 75/25 data split.

The following results are from the second dataset.

TABLE VI
STATISTICS OF ALGORITHMS WITH 75/25 DATA SPLIT

Model	Accuracy	Precision	Recall	f1-score	runtime
Logistic Regression	0.87	0.93	0.50	0.46	0.696
SVM	0.90	0.95	0.62	0.67	1.477
k-NN	0.87	0.93	0.51	0.47	1.752
Random Forest	0.90	0.95	0.62	0.67	2.712
Decision Tree	**0.90**	**0.95**	**0.62**	**0.67**	**0.94**
Naïve Bayes	0.87	0.93	0.50	0.46	0.357

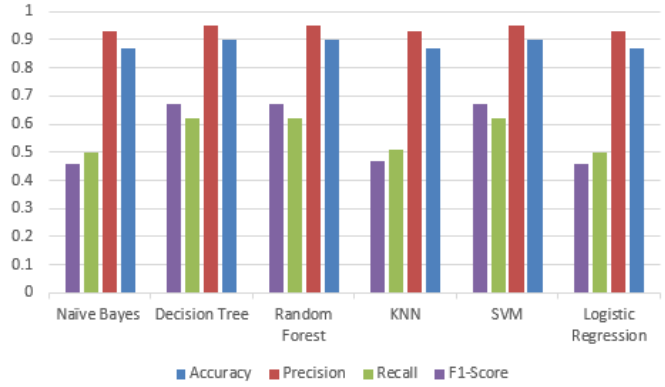

Fig. 7. Second dataset performance chart with data split

The second dataset had generally slightly more accuracy than the first dataset in all it's conditions. SVM, Random forest and Decision tree are sharing the same accuracy once again which is 0.90 in this case with Naive Bayes and K-NN having a slightly lower Recall.LR did not change at all,they were all relatively fast in building models except Random forest took slightly more time.

TABLE VII
STATISTICS OF ALGORITHMS WITH 5 K-FOLD

Model	Accuracy	Precision	Recall	f1-score	runtime
Logistic Regression	0.87	0.93	0.50	0.46	1.863
SVM	0.90	0.95	0.62	0.67	12.59
k-NN	0.87	0.93	0.52	0.49	1.435
Random Forest	0.90	0.95	0.62	0.67	21.381
Decision Tree	**0.90**	**0.95**	**0.62**	**0.67**	**5.281**
Naïve Bayes	0.87	0.93	0.50	0.47	0.813

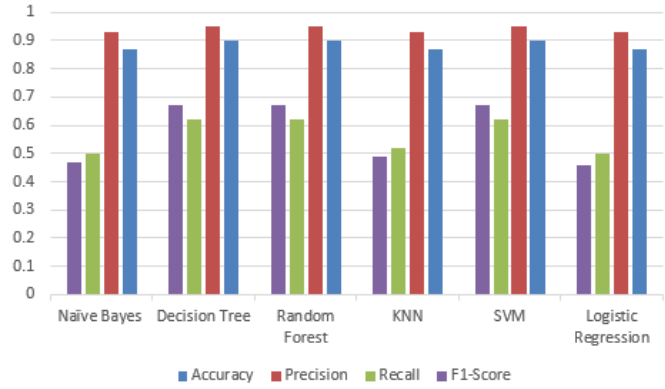

Fig. 8. Second dataset performance chart with 5 k-fold

The results using k-fold were very similar, maybe slightly better in f1-score than the 75/25 data split, and it also maintained the same hierarchy.

The following results are from the third dataset.

TABLE VIII
STATISTICS OF ALGORITHMS WITH 75/25 DATA SPLIT

Model	Accuracy	Precision	Recall	f1-score	runtime
Logistic Regression	0.87	0.93	0.50	0.46	0.801
SVM	**0.89**	**0.94**	**0.56**	**0.58**	**1.262**
k-NN	0.87	0.94	0.51	0.49	1.368
Random Forest	0.89	0.94	0.56	0.58	2.649
Decision Tree	0.89	0.94	0.56	0.58	1.369
Naïve Bayes	0.87	0.93	0.51	0.48	0.596

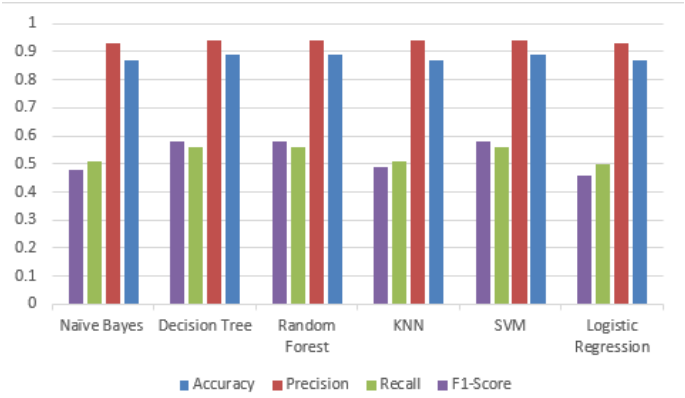

Fig. 9. Third dataset performance chart with data split

Surprisingly, no changes compared to the first dataset, which could be attributed to the similarity in numbers of rows in datasets. All the algorithms had the same results as the first time.

TABLE IX
STATISTICS OF ALGORITHMS WITH 5 K-FOLD

Model	Accuracy	Precision	Recall	f1-score	runtime
Logistic Regression	0.87	0.93	0.50	0.46	1.795
SVM	0.89	0.95	0.60	0.63	10.508
k-NN	0.87	0.94	0.52	0.51	3.311
Random Forest	0.89	0.95	0.60	0.63	22.999
Decision Tree	**0.89**	**0.95**	**0.60**	**0.63**	**5.455**
Naïve Bayes	0.87	0.93	0.50	0.47	1.274

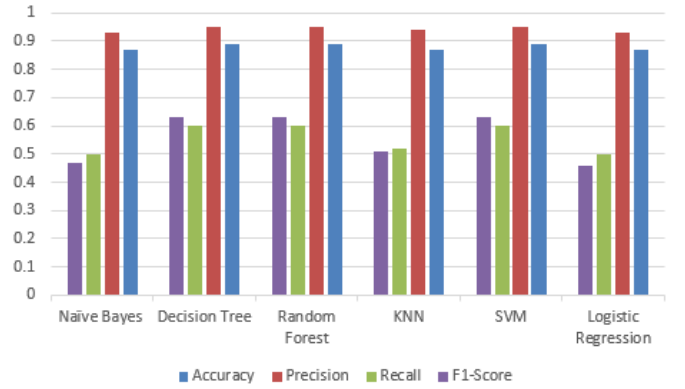

Fig. 10. Third dataset performance chart with 5 k-fold

Using k-fold, all f1-scores were slightly higher in all the algorithms.

V. Conclusion

Our research tackles the need for robust SMS spam detection, introducing a framework with six machine learning algorithms evaluated across three datasets. Decision Tree consistently excels with percentages of 89%, 90%, and 89%, justifying its choice over SVM and Random Forest for runtime efficiency. Our commitment to Decision Tree emphasizes both high accuracy and practical use. As technology advances, machine learning will enhance communication channel protection from spam, providing sophisticated tools for diverse challenges, supported by insights from three datasets.

VI. Acknowledgment

Initially, we extend our appreciation to the dedicated staff at Misr International University, particularly within the faculty of computer science, whose diligent efforts have contributed significantly to the success of this institution. We express our sincere gratitude to Prof. Mohamed Shebl El Komy, the University President, Prof. Dr. Ayman Nabil, Dean of the Faculty of Computer Science, and Prof. Abdelnasser Zaied, Vice Dean of Student Affairs and a Professor of Computer Engineering, for their roles in facilitating our learning experience within this esteemed university. Lastly, special recognition is owed to Dr. Diaa AbdelMoneim, an associate professor in information systems, and Eng. Tarek and Eng. Mohamed, for their invaluable support and guidance.

References

[1] F. Z. El Hlouli, J. Riffi, M. A. Mahraz, A. El Yahyaouy, and H. Tairi, "Detection of sms spam using machine-learning algorithms," in *Embedded Systems and Artificial Intelligence: Proceedings of ESAI 2019, Fez, Morocco*. Springer, 2020, pp. 429–440.

[2] A. Kipkebut, M. Thiga, and E. Okumu, "Machine learning sms spam detection model," 2019.

[3] L. GuangJun, S. Nazir, H. U. Khan, and A. U. Haq, "Spam detection approach for secure mobile message communication using machine learning algorithms," *Security and Communication Networks*, vol. 2020, pp. 1–6, 2020.

[4] M. R. Julis and S. Alagesan, "Spam detection in sms using machine learning through textmining," *International Journal Of Scientific & Technology Research*, vol. 9, no. 02, 2020.

[5] G. Sethi and V. Bhootna, "Sms spam filtering application using android," *Int. J. Comput. Sci. Inf. Technol*, vol. 5, no. 3, pp. 4624–4626, 2014.

[6] P. Poomka, W. Pongsena, N. Kerdprasop, and K. Kerdprasop, "Sms spam detection based on long short-term memory and gated recurrent unit," *International Journal of Future Computer and Communication*, vol. 8, no. 1, pp. 11–15, 2019.

[7] X. Liu, H. Lu, and A. Nayak, "A spam transformer model for sms spam detection," *IEEE Access*, vol. 9, pp. 80 253–80 263, 2021.

[8] S. Sheikhi, M. T. Kheirabadi, and A. Bazzazi, "An effective model for sms spam detection using content-based features and averaged neural network," *International Journal of Engineering*, vol. 33, no. 2, pp. 221–228, 2020. [Online]. Available: https://www.ije.ir/article_103370.html

[9] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A comparative study of spam sms detection using machine learning classifiers," in *2018 eleventh international conference on contemporary computing (IC3)*. IEEE, 2018, pp. 1–7.

[10] M. Popovac, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Convolutional neural network based sms spam detection," in *2018 26th Telecommunications forum (TELFOR)*. IEEE, 2018, pp. 1–4.

[11] H. Y. Aliza, K. A. Nagary, E. Ahmed, K. M. Puspita, K. A. Rimi, A. Khater, and F. Faisal, "A comparative analysis of sms spam detection employing machine learning methods," in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2022, pp. 916–922.

[12] S. K. Trivedi, "A study of machine learning classifiers for spam detection," in *2016 4th international symposium on computational and business intelligence (ISCBI)*. IEEE, 2016, pp. 176–180.

[13] P. Navaney, G. Dubey, and A. Rana, "Sms spam filtering using supervised machine learning algorithms," in *2018 8th international conference on cloud computing, data science & engineering (confluence)*. IEEE, 2018, pp. 43–48.

[14] C. Oswald, S. E. Simon, and A. Bhattacharya, "Spotspam: Intention analysis–driven sms spam detection using bert embeddings," *ACM Transactions on the Web (TWEB)*, vol. 16, no. 3, pp. 1–27, 2022.

[15] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection," *Procedia Computer Science*, vol. 190, pp. 479–486, 2021.

[16] T. B. Shahi, A. Yadav *et al.*, "Mobile sms spam filtering for nepali text using naïve bayesian and support vector machine," *International Journal of Intelligence Science*, vol. 4, no. 01, pp. 24–28, 2014.

[17] H. Shirani-Mehr, "Sms spam detection using machine learning approach," *unpublished) http://cs229. stanford. edu/proj2013/Shir aniMeh r-SMSSpamDetectionUsingMachineLearningApproach. pdf*, 2013.

[18] S. D. Gupta, S. Saha, and S. K. Das, "Sms spam detection using machine learning," in *Journal of Physics: Conference Series*, vol. 1797, no. 1. IOP Publishing, 2021, p. 012017.

[19] D. Suleiman and G. Al-Naymat, "Sms spam detection using h2o framework," *Procedia computer science*, vol. 113, pp. 154–161, 2017.

[20] K. SUPARNA and V. V. T. S. VARMA, "Sms spam detection using machine learning," *Journal of Engineering*

Sciences, vol. 14, no. 09, 2023.

[21] [Online]. Available: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/code?datasetId=483&sortBy=voteCount

[22] [Online]. Available: https://www.kaggle.com/datasets/vivekchutke/spam-ham-sms-dataset/data

[23] [Online]. Available: https://www.kaggle.com/datasets/mrushan3/spam-sms-classification?select=TrainDataset.csv

[24] [Online]. Available: https://scikit-learn.org/stable/modules/svm.html#:~:text=Support%20vector%20machines%20(SVMs)%20are,Effective%20in%20high%20dimensional%20spaces.

[25] M. Hameed, M. Hassaballah, M. Hosney, and A. Alqahtani, "An ai-enabled internet of things based autism care system for improving cognitive ability of children with autism spectrum disorders," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–12, 05 2022.

[26] [Online]. Available: https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes.

[27] [Online]. Available: https://www.ibm.com/topics/naive-bayes#:~:text=the%20next%20step-,Na%C3%AFve%20Bayes%20classifiers,a%20given%20class%20or%20category.

[28] [Online]. Available: https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.

[29] [Online]. Available: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.

[30] [Online]. Available: https://www.ibm.com/topics/logistic-regression