Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 5, 2025

SMART FILTERS FOR SMS SPAM: A MACHINE LEARNING APPROACH TO SMS CLASSIFICATION

Ishrat Nawaz[1], Saima Noreen Khosa[2], Rida Fatima[3], Muhammad Saeed[4], Muhammad Shadab Alam Hashmi[*5]

*[1,2,3,*5]Institute of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan*
[4]Department of Computer Science, Govt. Graduate College Bahawalnagar, Pakistan

[1]writetoishratnawaz@gmail.com, [2]saimakhosa@yahoo.com, [3]fatima.rida55@gmail.com, [4]saeed.ggcbwn@gmail.com, [*5]shadab.alam@kfueit.edu.pk

DOI: https://doi.org/10.5281/zenodo.15333801

Keywords
Machine Learning; SMS; Spam Detection ; SMS spam collection.

Article History
Received on 25 March 2025
Accepted on 25 April 2025
Published on 03 May 2025

Corresponding Author: * Muhammad Shadab Alam Hashmi

Abstract

The exponential rise in the number of undesired text messages delivered via SMS has been directly related to the explosion in the number of mobile phones sold. Although various information channels are considered "spotless" and trustworthy in many parts of the world, ongoing reports show that cell phone spam is significantly increasing. It is a big problem. It is becoming increasingly pervasive worldwide, especially in Asia and the Middle East. In the same way that finding a solution to such an issue can be time-consuming, so can the process of identifying spam texts from genuine communications. It solves many difficulties and makes life much easier because it can distinguish between real SMS and spam. In any event, it faces specific challenges and obstacles that are unique to itself. During this current research, we have investigated five Machine Learning (ML) methods to identify spam in a short text message using a single dataset containing SMS spam Collection. The SMS spam dataset was extracted from the Kaggle repository. The experiment is carried out on the R platform. Eleven characteristics, including binary and numeric features like Char Count, Has number, Has URL, Has Date, Has dollar, Emoticon, Email, and Phone, as well as spam count, ham count, and spam binary, are employed in this research. These features are used for feature selection and showing results using Machine Learning(ML) approaches. The effectiveness of the various strategies or methods is evaluated using metrics such as sensitivity, accuracy, precision, F1 score, recall, and specificity. The outcomes show that the light gradient boosting machine (LGBM) with these features achieved a sensitivity score of 100, precision score of 100, F1 score of 100, recall of 100, and specificity score of 100, with an optimal accuracy score of 100 percent, which is outstanding compared to all other state-of-the-art studies.

INTRODUCTION

Thanks to an extensively utilized text messaging data protocol known as Short Messaging Service (SMS), mobile phone devices can communicate with one another to send and receive brief text messages through industry-standard protocols. Over the previous ten years, researchers observed a growth in the total number of SMS text messages sent due to the proliferation of mobile phone use. Around the world, there are currently 5.31 billion unique users using mobile phones, according to the most recent numbers released by GSMA intelligence[1]. It represents an increase of 95 million from the previous year. About seventy five percent of the global population, or 500 million peoples sends and receives SMS messages. When SMS spam originally started, it frequently only included business advertisements. Contemporary spam today commonly contains attachments or website links tainted with malware and spyware, turning the user into a covert target of numerous cybercrimes.

SMS spam is a significant issue on Facebook, WhatsApp, and other social messaging services. Many people still expose their passwords, account numbers, and other personal information when they respond to spam mailings. Even though the networks of these social platforms can identify and prevent spam messages, SMS still faces serious spam issues. Therefore, problems caused by spam could range from irrelevant bothers to serious security problems. Spamming has been a major issue in Far Eastern nations since 2001. Spam accounted for more than 66 percent of all Internet-based SMS messages transmitted by 2005 and grew to 70 percent by 2010 [1]. However, in 2015, it increased to 73 percent, which is significant, whereas it has essentially frighteningly increased to 85 percent of all SMS. Considered to be a more serious social media issue is SMS spam. As a result of the fact that numerous customers are not yet knowledgeable about insurance components, their mobile devices are susceptible to digital assaults. The government has established the NCPR (National Customer Preference Register) library, which has helped reduce spam calls to some extent but does not filter spam SMS. Although various datasets are available to test methods for recognizing spam in email, the datasets that can be used to build and test procedures for recognizing spam in SMS still need to be revised, and their estimates need to be more accurate [2].

Spam will typically contain doubtful links intended to harm the user's data and the many methods the spammer uses to gather the user's email address, such as through chat rooms, news groups, websites, and other online forums. Spammers use many different methods, such as appending, picture spam, black spam, and back scatter spam, among many others. The following Figure 1 illustrates the architecture behind detecting spam in SMS messages.

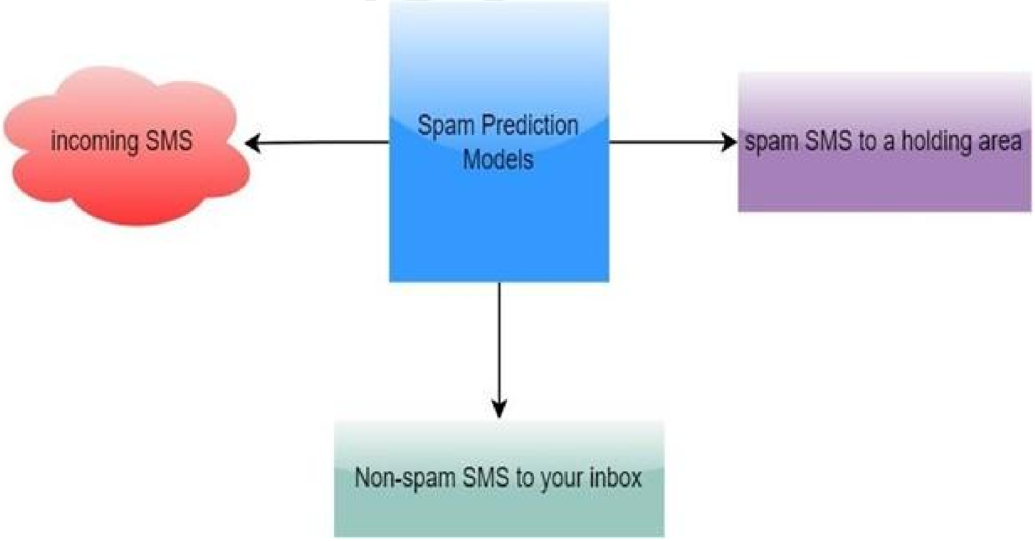


Figure 1: Architecture of SMS spam detection

In earlier studies, many approaches and models were utilized to detect spam. These include using unsupervised and supervised learning machines, such as the SVM and the H20 framework[3]. A framework called semi-supervised spam detection (S3D) or a decision tree algorithm [4] long short-term memory and K-Nearest neighbor[5], naive Bayes (NB), Deep Neural Network (DNN), Decision tree, Artificial Neural Network (ANN), Random forest, and collaboration-based and content-based method[6] these are ML techniques that have been devised.

Various strategies that use a variety of data sets were utilized to carry out individual comparisons and tests. Different datasets in this research permit a complete knowledge of the topic, as each dataset yields different findings. According to these findings, the accuracy can be achieved using classifier modules.

The main objective of our study is to analyze the use of ML strategies to identify SMS spam sent by text message. The primary goal is to distribute the comprehensive findings of this application, which can be used to determine whether a message is spam or harmful. These issues can be effectively addressed by utilizing machine learning methods, which form the basis of the newly proposed SMS spam detection system. The dataset, named the SMS Spam Collection Dataset, was sourced from Kaggle and is publicly available. This study employs a range of natural language processing methods to thoroughly investigate the impact of performance on the purpose model. Various machine learning algorithms, including Naïve Bayes (NB), Gradient Boosting Machine (GBM), Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR), are used in this research. Machine Learning (ML) algorithms in this study work with both Binary and numeric features. The 11 features used in this research, such as Char Count, Number count, Ham count, and Spam count, are numeric features, while number, URL, Date, Dollar, Emoticon, Email, and Phone are binary features. This study has led us to explore the answers to these research problems, providing a thorough understanding of the issue.

- Do extracted binary features have the ability to enhance the accurateness of SMS spam prediction?
- Do the extracted numeric features hold the potential in SMS spam prediction?
- Are frequent words in spam and ham SMS valid as feature sets for spam prediction?
- Which machine learning algorithm could achieve the best accuracy.

1. Related Work

In this section, various studies related to SMS and Twitter spam detection using machine learning (ML) and deep learning (DL) models are reviewed. The key models, datasets, and results are summarized below in the Table.

[3] The author of this study conducted an experiment on spam identification using the H2O learning model. The H2O context presents various machine learning (ML) libraries, where the H2O stage serves as an open-source framework. Machine learning methods DL, RF, and NB were developed on H2O and were utilized to recover the spam detection process. These algorithms were named after their respective building blocks. According to the author, the experiments were carried out utilizing the H2O stage, and the data that was employed came from several sources associated with UCI Machine Learning. DL and RL are used in classification to describe the main elements of detecting SMS spam. Based on the results of the studies, the naive Bayes (NB) classifier dataset consists of 5572 messages, and its accuracy is 97.7 percent accuracy score, precision score 96, 86 percent recall score, and 91 percent f-measure score.

The authors[7] of this study utilize two different benchmark datasets, where the first is an SMS spam corpus and the second is a Twitter corpus. Both datasets each receive their unique version of the dictionary of words. Other datasets, such as Twitter's, comprise 97,831 words, with 14538 unique terms. The SMS dataset has 85,477 words in total, with 8,277 unique words. On the SMS Spam dataset, the accuracy of the learning model is 97.5 percent, however on the Twitter dataset it give only 93.43 percent.

The researcher of this paper[8] presented Convolutional Neural Networks (CNNs) and long-term neural network-based deep learning architectures. CNN and LSTM were able to preserve their accuracy by presenting the evidence in the form of symbol words based on Concept Net. The Scholar of this study used two datasets: SMS spam and the Twitter dataset, respectively; their respective accuracy levels are 98.86 percent and 95.88 percent.

According to the authors of this study [9], they attempted to use LSTM and CNN techniques to detect spam messages sent by SMS. The authors assessed the effectiveness of the LSTM and CNN by contrasting their results with those obtained from the NB, Stochastic Gradient Descent (SGD), RF, Gradient Boosting (GB), and LR models. When detecting spam messages sent via text message, the exploratory outcomes indicated that LSTM and

CNN surpassed classic machine learning classifiers that were put to the test and attained a 99.44 score of accuracy.

In this study,[10] The authors presented a variation of LSTM that included an additional semantic layer. This memory was called the SLSTM (Semantic Long Short-Term Memory). The datasets used in this study were the Twitter dataset and the SMS spam dataset; The authors integrated ConceptNet, WordNet, and Word2vec as the semantic layer for the identification classifier for SMS spam and used the SLSTM in conjunction with it. According to the results, the SLSTM model obtained an accuracy of 98.74 percent and 95.54 percent in SMS spam and Twitter datasets when applied to a dataset of SMS spam.

In this research study,[11] the authors proposed an approach to detecting spam from SMS, namely (spam Transformer), which was tested on UtkMl's Twitter and SMS Spam datasets. These datasets were evaluated against the benchmark advanced ML techniques. The suggested model exhibits good results with an accuracy score of 87.06 percent in UtkMl's Twitter dataset, which indicates a promising possibility of applying the model to comparable situations. The results of our research on the identification of SMS spam Illustrate that the recommended technique enhanced the spam Transformer approach gained highest F1-Score, 96.13, recall score of 0.9451, precision score of 97.81, and accuracy of 98.92. It is a comparison to all of the other potential choices.

This study[12] suggests an approach for identifying spam communications via analysis of the emotional tone of the textual data in the email's body. To investigate the emotional and chronological aspects of texts, we use Word Embeddings and a Bidirectional LSTM network. In addition, by employing a Convolution Neural Network, we could shorten the time needed for training and get higher-level text characteristics for the bidirectional LSTM network. The performance of their Project approach was compared and evaluated in two datasets, namely the SMS spam and ling spam datasets, and they applied recall, precision, and f-score metrics. The improved performance of our model yields an accuracy of approximately 98.30 percent in the SMS spam dataset and a 98 percent accuracy score produced on the ling spam dataset by the Bi-LSTM model. In addition, the author shows that our proposed model beats well-known classifiers and the most advanced strategies currently available for specifying spam transmissions, revealing the distinction of our method alone.

The authors of this research[13] have experience with the daily use of mobile SMS, a service available on smartphones, and SMS spam traffic; spammers use different methods to make spam possible, like lottery tickets, credit card information, etc. SMS spam has also increased drastically, so SMS spam classification requires special attention. The researcher uses UCI's publicly available dataset to build SMS spam identification, and various ML and DL models were utilised. Our exploratory outcomes have displayed that the LSTM algorithm surpasses prior algorithms in spam identification with an accurateness score of 98.5. The state-of-the-art study of Python programming is employed to make all outcomes possible.

[14]The researcher presents a unique technique to detect and filter spam communications in this study. The author delivers a unique strategy "Term Frequency— Inverse Document Frequency (TF-IDF)" variation with Multinomial Naïve Bayes (MNB) and Support Vector Machine (SVM) machine learning classifiers. The metrics used for this study to measure the score were accuracy, F1-score, and computational time. The 98.50 Accuracy score was attained through experimental examination, 0.98 F1-Score under the roc curve (AUC) of 0.97 score for the MNB classifier with TF-IDF after stemming.

The Researcher in this approach[15] presented a unique ML approach to identifying spam SMS messages using feature selection decision-making. To lower the complexity and enhance the classifier performance by extracting related attributes from the dataset. In the second phase, a neural network model was employed to extract attributes to categorize the messages into spam or legitimate classes. The evaluation is done on the base accuracy and F-measure metrics, which reached satisfactory accuracy on a real-world dataset of 5000 messages. We attained an adequate level of accuracy by employing Recurrent Neural Networks for SMS spam identification. In this study, the results were compared with those of previous studies, and this approach achieved state-of-the-art high detection results in terms of F1-measure, precision, recall, and classification accuracy compared with other considered research. The suggested Hybrid model attained a 0.988 accuracy, 0.9892 precision, 0.9929 F-1 and 0.9967 recall score.

Table 1: Literature Review of Previous Work

References	Year	Dataset	Proposed Technique	Accuracy
[3]	2020	UCI Machine Learning Repositories	RF, ML	97.7
[7]	2019	2019 & SMS spam corpus and Twitter corpus	Hybrid model CNN, LSTM, DL	97.5 / 93.43
[8]	2019	SMS Spam & Twitter	SSCL, DL	98.86 / 95.88
[9]	2020	2020 & Text Based Dataset	CNN, DL	99.44
[10]	2019	SMS Spam Collection dataset and Twitter dataset	LSTM, DL	98.74 / 95.54
[11]	2021	SMS spam & UtkMl's Twitter dataset	Spam Transformer, DL	98.92
[12]	2020	SMS spam dataset and Ling spam dataset	Bi-LSTM, DL	98.3 / 98
[13]	2021	UCI SMS Spam dataset	LSTM, DL	98.5
[14]	2023	UCI SMS Dataset	Multinomial Naive Bayes (MNB), ML	98.50
[15]	2020	UCI Machine Learning Repository	Hybrid Model, ML	98.8

2. Proposed Methodology

A comprehensive literature on SMS spam detection shows that the researchers have produced a wide range of research articles using machine learning algorithms and textual features of SMS data, i.e., TF-IDF (term frequency-inverse document frequency)[16] and BOW (bag of words)[17]. To our knowledge, no classification study relies entirely on freely available binary and numeric metadata. This study focuses on binary and numeric classification via machine learning algorithms: given an SMS, classify it as spam or ham, exploring its binary and numeric features. The main aim of this study is to recognize SMS as spam or ham by examining the importance of ML classifiers. Specifically, this study will look into how these impacts are brought about. Any unwanted communication delivered from a mobile phone, such as text messages over the Short Message Service (SMS), is called SMS spam or cell spam. SMS ham is a magnificent message.

The SMS spam classification problem is solved as described in the section. The Methodology diagram is shown in Figure 2.

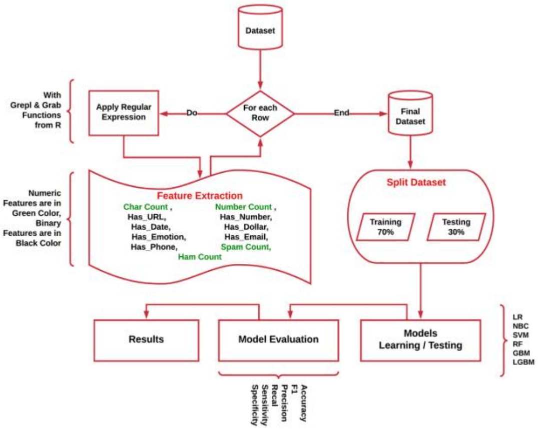


Figure 2 : Architecture of the Proposed Methodology

Binary features (i.e., has url, has date, has emotion, has phone, has number, has dollar and has email), as well as numeric features (i.e., char count, number count, spam count and ham count) have been utilize by the set of the ML classifiers in our research i.e., LR and NBC[18], SVM[19], RF[20], GBM and LGBM[21]. After the feature engineering extraction was completed, the dataset was subdivided into two unique sets: the first set was reserved for use in the training process. In contrast, the second set was reserved for use in the testing process. Following the development of the features, we divide the dataset into two distinct sets or ways: the training set accounts for seventy percent of the total, while the testing set represents the remaining thirty percent. After training the learning model with the help of the training set, we evaluate how well the model learnt by giving it test data after training. This helps us determine how well the model learned. Furthermore, to check the accuracy of machine learning models, the results have been evaluated using a set of estimated metrics.(accuracy, sensitivity, specificity, precision, recall and F1-score). Details about each step used in SMS spam detection are given in Figure 2.

The dataset includes spam text messages that are available to the general public and may be in the Kaggle Repository[22]. The collection of SMS-branded messages is the sms spam array, which was created to research SMS spam. This collection was created to study SMS Spam. It comes with a database of 5,574 English SMS messages labelled as either ham (genuine) messages 4825 and spam SMS found 747, depending on the sender's intent.

The SMS Spam Collection is a set of SMS-tagged messages that have been collected for SMS Spam research. It consists of a CSV file or document containing 5,574 English SMS messages, each identified as either ham (genuine) or spam, and two v1 and v2 attributes. The v2 represents the data messages, all of which either contain spam or do not contain spam. The much-anticipated mark iv1 is divided into two categories: i0, which stands for "non-spam," and 1, for "spam." In the data, there were 4825 ham tests and 747 spam tests. Figure 3 summarises the count dataset utilized in the inquiry and the attribute modifications applied to it. These characteristics were utilized in either this analysis or the analysis of ham and spam. Figure 4 represents the top 10 frequent words in spam and spam messages.

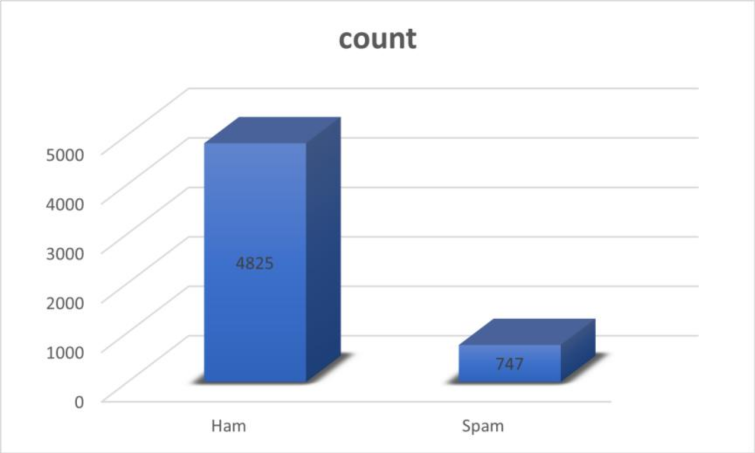


Figure 3 : Dataset Description

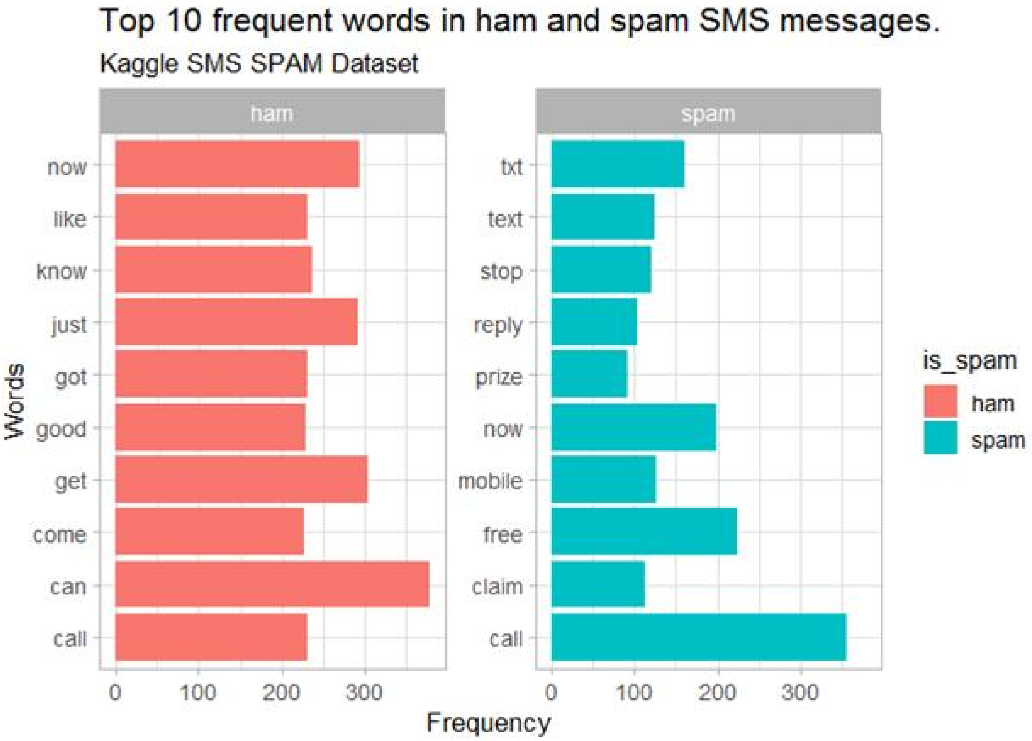


Figure 4 : Top 10 Frequent Words in Ham and Spam Messages

3.1 Preprocessing

A dataset's features can be improved by applying preprocessing after cleaning the dataset. The dataset will be made accessible from noise using the preprocessing approach, which will repair spelling errors, reduce the number of repeated characters, and disambiguate ambiguous abbreviations. In addition, using preprocessing techniques for SMS spam detection, such as stop words, removal of punctuation, word stemming, tokenization, number removal, and conversion to lowercase, increase the consistency of the dataset. In the ongoing research and study effort, some preprocessing methods are utilized to clean the data. The structure of preprocessing is shown in figure 5.

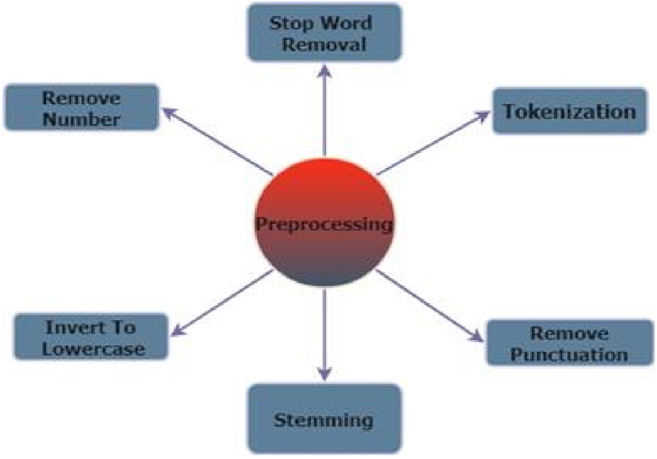


Figure 5 : Structure of Preprocessing

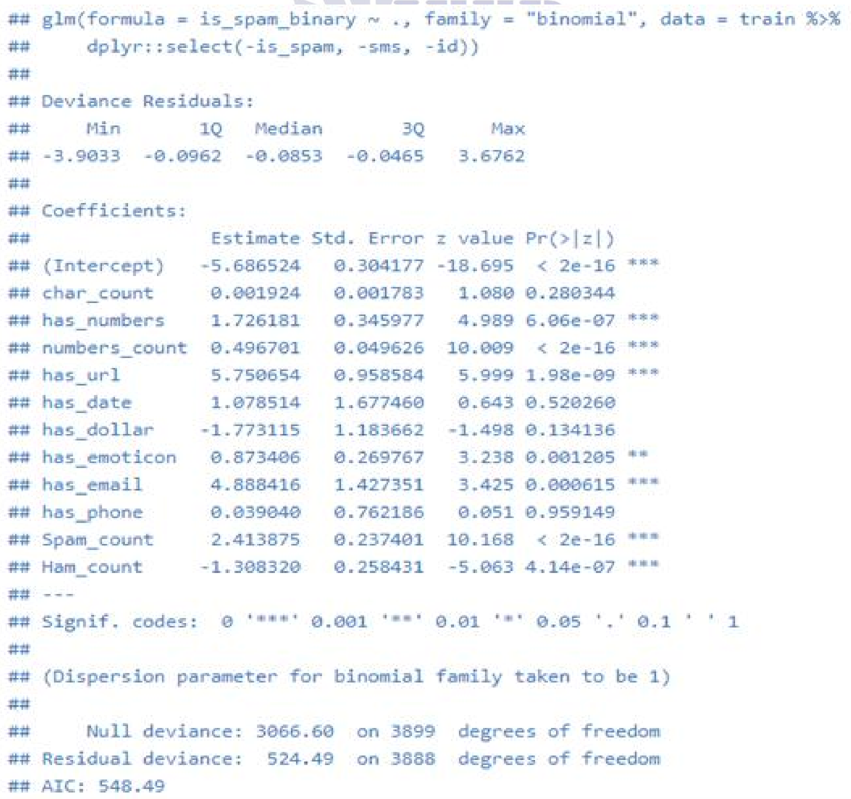
3.2 Feature Extraction

In the dataset from Kaggle, we applied different preprocessing (i.e., stop word and punctuation removal) steps. As we need to extract the metadata about each SMS, we regularly use expressions on SMS text with the help of functions (i.e., grab and grip) provided in the R tool. In this way, we extracted 11 features from the dataset, and the details about each feature are presented in Table 2 and illustration detail of extracted features is presented in figure 6.

Eleven highlights or features we have eliminated and tested for our proposed system include Char Count, Has number, Has URL, Has Date, Has dollar, Has Emoticon, Has Email, Has Phone, Spam count, Ham count, and number count. Char count, Number count, Spam count, and Ham count are numeric features. Has number, Has URL, Has Date, Has dollar, Has Emoticon, Has Email, and Has Phone are binary features. These features are addressed for text grouping or classification in this current study.

Table 2 :Extracted Independent Variables for Building Models

Feature Name	Description	Type
Char count	Number of characters in message	Numeric
Has number	Whether message contains a number (e.g., 23, 45, 6, 34)	Logical
Has URL	Whether message contains a URL (e.g., www.google.com)	Logical
Has date	Whether message contains a date (e.g., 1/03/2007)	Logical
Has dollar	Whether message contains a dollar sign (e.g., $)	Logical
Has emoticon	Whether message contains emotions (e.g., Sad, Happy)	Logical
Has email	Whether message contains an email (e.g., xyz@gmail.com)	Logical
Has phone	Whether message contains a phone number (e.g., 0322xxxxxxx)	Logical
Spam count	Number of words from the top 10 frequent spam words	Numeric
Ham count	Number of words from the top 10 frequent ham words	Numeric
Is-spam-binary	Class label (i.e., whether message is ham or spam)	Logical



```
## glm(formula = is_spam_binary ~ ., family = "binomial", data = train %>%
##     dplyr::select(-is_spam, -sms, -id))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9033  -0.0962  -0.0853  -0.0465   3.6762
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.686524   0.304177 -18.695  < 2e-16 ***
## char_count     0.001924   0.001783   1.080 0.280344
## has_numbers    1.726181   0.345977   4.989 6.06e-07 ***
## numbers_count  0.496701   0.049626  10.009  < 2e-16 ***
## has_url        5.750654   0.958584   5.999 1.98e-09 ***
## has_date       1.078514   1.677460   0.643 0.520260
## has_dollar    -1.773115   1.183662  -1.498 0.134136
## has_emoticon   0.873406   0.269767   3.238 0.001205 **
## has_email      4.888416   1.427351   3.425 0.000615 ***
## has_phone      0.039040   0.762186   0.051 0.959149
## Spam_count     2.413875   0.237401  10.168  < 2e-16 ***
## Ham_count     -1.308320   0.258431  -5.063 4.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3066.60  on 3899  degrees of freedom
## Residual deviance:  524.49  on 3888  degrees of freedom
## AIC: 548.49
```

Figure 6 : Analysis of Optimal Extracted Features

3.3 : Data Splitting

The dataset splitting is applied to the training and testing phases. For train and test purpose, we split data into an 80:20 ratio. To attain the desirable outcomes through advanced ML classifiers we employed, 20 percent unseen data was used on ML classifiers for prediction and 80 percent for the ML classifiers' training.

3.4 : Employed Machine Learning Models

Various ML classifiers are used to categorize SMS spam identification. Our research employed six advanced, well-trained, and established ML classifiers to predict the maximum SMS spam prediction outcome for SMS spam classification.

3.5.1 : Logistic Regression

The classifier used for categorization or predictive analysis is statistical logistic regression(LR) [23] . This classifier's primary objective is to estimate the possibility of an event occurring. For example, SMS is ham or spam is predicted by this classifier on the basis of independent variables in the dataset. We have predicted the SMS class (either spam or ham) based on SMS metadata (extracted binary and numeric features), with 'is-spam-binary' as the dependent variable and the remaining features as independent variables. One of the applications of this tool is examining binary data, in which one or more variables are utilized to catch the result. Logistic regression describes data and explains the link between one paired ward variable and at least one free component of the ostensible, ordinal, period, or proportion level. Logistic regression can analyze data at the ostensible, ordinal, period, or percentage level. This learning paradigm is the most effective when all students in the target class are absent. The cooperation between the all-out ward vector and at least one free factor is determined by applying a sigmoid LR skill to approximate the relevant probabilities. The sigmoid function is used to make predictions about the values of probabilities. The sigmoid function translates the values between zero and one [24].

3.5.2 : Naive Bayes(NB)

The Naive Bayes (NB) classifier, a machine learning approach with extensive experience, is considered among the best and is widely popular. Its effectiveness, while still retaining its ease of use, is the critical factor contributing to its widespread popularity. This probabilistic classifier characterizes information occurrence based on the likelihood of features, making it a powerful tool for data analysis. It also predicts whether SMS is spam or ham based on independent variables in the given dataset. We have predicted the SMS class (spam or ham) based on SMS metadata (extracted binary and numeric features). Where' is spam-binary' is used as a dependent variable, the remaining features are used as independent variables. The NB classifier is a managed statistical learning algorithm based on Thomas Bayes's formulation of Bayes' Theorem [25].

3.5.3 : Random Forest (RF) :

It is a supervised classifier mainly used in the categorization task popular and commonly used algorithm that is helpful for regression problems. It builds decision trees via various samples using the average votes for regression and the majority votes for classification. Using this model, we have predicted the SMS class (i.e., either spam or ham) using SMS metadata (i.e., extracted binary and numeric features). Where' is-spam-binary' is used as a dependent variable, the remaining features are used as independent variables. A decision tree is built for each tree by considering a random subset of attributes for each decision hub in the tree. Consider the following illustration of a decision tree:" standardized TF/IDF rate for token"limit. The Random Forest method combines selecting individual elements with considering many element subsets. (Rather than zeroing in on only a couple, which includes the best independent preparation information). The crucial bounds of the random forest model consist of multiple attributes and trees that must be built [26]. Also RF algorithm introduces randomness during both feature selection and dataset bootstrapping, which promotes diversity among the constituent trees that can generalize new unseen data more effectively and helps prevent overfitting [27].

3.5.4 : Support Vector Machine :

It is a classifier famous for categorization, regression, and outlier detection. The main thing about SVM is that it is effective in high-dimensional spaces, i.e., when we have several dimensions greater than samples. Here, we used this model to predict SMS class (i.e., either spam or ham) using available SMS metadata (i.e., extracted binary and numeric features). Where is spam-binary' is used as a dependent variable, the remaining features are used as independent variables [28]. A non-probabilistic Support Vector Machine (SVM) is a controlled learning classifier that assigns class names to test data. The preparatory text is an N-dimensional vector, a list of numbers foci in an N-dimensional space. SVM efficiently locates an appropriate hyperplane of size n minus one that separates the various groups of information objects. While there may be many hyperplanes that can serve as classifiers, SVM selects the one that maximizes the distance between the classes on each side of the data points. Considering both the benefits and drawbacks of using SVM for Data Analysis is essential because it uses kernel functions to separate data points in an n-dimensional space and identifies the optimal boundary between classes [29].

3.5.5 : Gradient Boosting Machine (GBM) :

It is a classifier commonly utilised to train weak classifiers into robust learning classifiers. In GBM, every weak learning classifier is in the updated interpretation of the foremost dataset. Decision trees are frequently utilized during angle increasing. "gradient boosting" refers to taking a weak learning estimate or a poor speculation and causing a progression of adjustments. These changes will ultimately result in the hypothesis having greater validity. The comprehension of probability approximately correct learning is the primary subject of this variation of the boosting hypothesis. (PAC). There are some positives and negatives regarding the Gradient Boosting Machine [30]. Here, we used this model to predict SMS class (i.e., either spam or ham) using available SMS metadata (i.e., extracted binary and numeric features). Where is_spam_binary' is used as a dependent variable, and the remaining features are used as independent variables.

3.5.6 : Light Gradient Boosting Machine (LGBM) :

The Train Using Auto ML tool implements a gradient-boosting ensemble method known as Light GBM. This method is founded on decision trees and is employed by the utility. Light GBM is a decision tree-based method that may be used for classification and regression, similar to other tree-based methods. Light GBM is designed to provide excellent performance with distributed systems and has been optimized for this purpose. Two cutting-edge methods, referred to respectively as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling, are incorporated into Light GBM's implementation of a conventional Gradient Boosting Decision Tree (GBDT) algorithm. These techniques are Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling. Light GBM also makes use of exclusive feature bundles. (EFB). These methods are intended to make a significant contribution toward improving the GBDT's efficiency as well as its scalability [31].

4- Outcome and Discussion

This section investigates the experimental strategy and outputs to demonstrate the probability of SPAM or HAM for SMS categorization. The results, including complete attributes, are shown with a binary categorization task using the SPAM characteristic, in case SMS is spam or not is determined through a key spam indicator.

The data is then meticulously utilized to train the advanced ML classifiers. The measuring metrics used for assessment are accuracy, precision, recall, and F1-Score, which are thorough and comprehensive, instilling confidence in the results.

4.1: Experimental Design

The machine we used for the experiment contains a graphical processing unit (GPU) from HP i3 with 4 gigabytes of random access memory (RAM) installed on a computer with Intel i3 cores operating at a frequency of 3.2 gigahertz and Windows 10.

The dataset retrieved from KAGGLE is used for the experiments. Initially, the dataset contains 5574 observations with two features (such as SMS and class label). SMS represents the text message, and the class label represents either spessag message e. a Out message of 5574 SMS, there were 747 spam SMS, and the rest of the SMS were ham. First, we applied a loop throughout the dataset and used regular expressions on each SMS to extract the metadata (i.e., binary and numeric features). In this process, we extracted nine features: two were numeric, and seven were binary.

Furthermore, we found the top 10 most frequent words within each class, i.e., spam and ham. Here, the frequency means the number of times a word occurs in a collection. These frequent words led us to extract two more numeric features from the dataset. These features are known as ham count and spam count, where ham count represents the count of words that belong to the top 10 frequent ham words.

We divide the final dataset into training and testing phases in a 70:30 ratio, using a seed value of 123. The training set contains a randomly selected 70 percent of the observations, while the remaining 30 percent are used for testing. The seed value ensures that the same training and testing sets are selected each time the code is executed. We then train different machine learning models using the training set and validate them on the testing set. Finally, we evaluate these models using state-of-the-art evaluation metrics, including sensitivity, specificity, accuracy, precision, recall, and F1 score.

4.2 : Evaluation Parameters

To assess their performance, we use a set of evaluation measures, including accuracy, precision, recall, F1-score, sensitivity, and specificity. These metrics are employed to evaluate the effectiveness of the proposed spam detection classifier in this research. The score for each measure is derived from the confusion matrix, as shown in Table 3 below.

Table 3: Confusion Matrix

Type	Spam	Ham
Spam	FP	TP
Ham	TN	FN

- True Positive (TP): The model correctly predicted that the SMS was Spam.
- True Negative (TN): It is a type 1 error. The model predicted the SMS was Spam, but it was not Spam.
- False Positive (FP): The model incorrectly predicted that the SMS was not Spam, while it was actually Ham.
- False Negative (FN): It is a type 2 error; the model predicted the SMS was not Spam, but it was Spam.

Through this confusion matrix, several evaluation scores are defined as below:

4.2.1 : Accuracy

The accuracy score of a message can be expressed either as a fraction or percentage of the total messages, indicating the model's overall performance. After that, multiply that number by 100 to get the percentage. It is intended as Equation 1.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

4.2.2 : Precision Score

In the context of spam detection, the term 'precision' refers to the percentage of spam communications correctly identified as such. This metric is a key factor in assessing the effectiveness of a spam filter. It is intended as Equation 2.

$$\text{Precision-score} = \frac{TP}{TP + FP}$$

4.2.3 : Recall Score

In the context of machine learning, the recall score, which is the total number of false positives for spam messages, is a critical metric. It is calculated by first separating the number of genuine positives from the combined sum of genuine and false negatives. A low recall number indicates a high number of erroneous negative results. It is meant to be used as Equation 3.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4.2.4 : F1 Score

The F1 score is a comprehensive metric that combines accuracy and evaluation into one. It takes into account both accuracy and evaluation, thereby providing a more holistic view of the model's performance. By using F-measure and precision esteem, the F1 score improves grouping. The F1 score ranges from 0 to 1, with 0 representing the most pessimistic scenario and 1 representing the most favorable option. This comprehensive metric is intended as Equation 4.

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

4.2.5 : Sensitivity

The degree to which a machine learning model can distinguish good examples is referred to as its level of sensitivity. This statistic is sometimes called the "true positive rate," or

recall, depending on the context. When analyzing the efficacy of a model, one component that is considered is the model's sensitivity. This is because sensitivity allows us to see how many positive situations the model accurately detected. It is meant to be used as Equation 5.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

4.2.6 : Specificity

When evaluating the performance of a model based on its sensitivity, it is common practice to compare and contrast sensitivity and specificity. Specificity is the proportion of cases in which the model successfully rules out false positives. This means that a further percentage of true negatives deemed positive and could be referred to as false positives will be presented as positive results in the study. This percentage could also be described as a True Negative Rate (TNR), another possible term. When put together, specificity (the rate of actual negative results) and the rate of false positive results would always equal one. A model with a low specificity will incorrectly classify a significant proportion of negative findings as positive. In contrast, a model with a high specificity will accurately detect the vast majority of the negative results.it is represented as Equation 6.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

4.3 :Evaluation of Various Machine Learning Classifier Outcomes

In the results, we produced a confusion matrix with functional parameters to validate the model. This figure of the confusion matrix represents the actual and predicted values, and this matrix helps interpret different aspects of the classifier's quality. The confusion matrix is presented in Figure 7. In this Figure, at the start, we have a confusion matrix containing 1435 True Negative values out of (1435+11) 1446 actual negative values, and 200 True positive values out of (200+26) 226 actual positive values, which shows how efficiently the LR model is working. The accuracy score, 97.79, can justify the previous statement, and the model's error rate is 100 - Accuracy, 2.21 percent. The accuracy represents the set of correctly classified cases. We can see that there were 1672 cases; out of them, 1435 were correctly classified as 0, and 200 were correctly classified as 1. Thus, out of 1672 classifications, we have 1635 correct classifications, which are 97.7 percent.

Furthermore, the kappa measure in the confusion matrix is another statistical method to test inter-rater reliability and has been frequently used in the literature. The range of the kappa value starts from -1 and ends at +1, where 0 corresponds to an agreement expected by a random chance, and one corresponds to a perfect agreement. The author of the Kappa measure suggested that no agreement can be possible at a value less than 0, and more agreements are as follows:

- Slight: within a range of 0.01 - 0.20,
- Fair: within a range of 0.21 - 0.40,
- Moderator: within a range of 0.41 - 0.60,
- Substantial: within a range of 0.61 - 0.81,
- Perfect: within a range of 0.81 - 1.00.

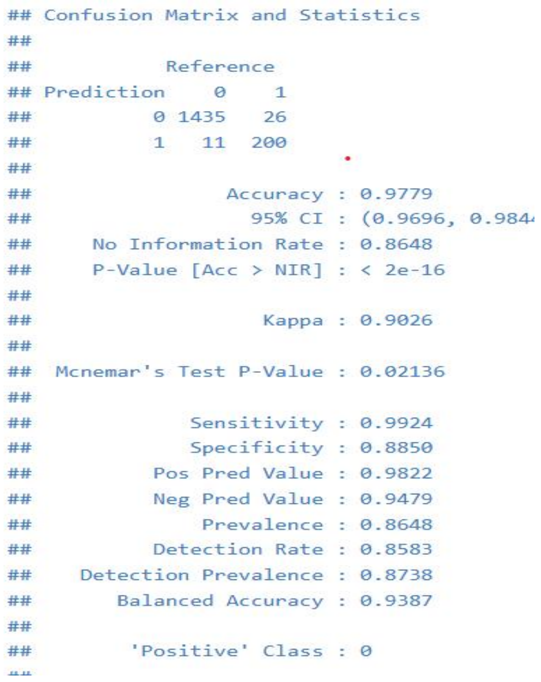
The confusion matrix reveals a perfect agreement in our model's predictions. This level of agreement is significant, as it aligns with the commonly accepted 80 percent agreement rate. Therefore, as indicated by the kappa, our model's performance is considered acceptable.

Apart from this, other statistical parameters are also given in the confusion matrix. Furthermore, if we look at the p-value, it shows that the model's accuracy is better. In addition, the model's sensitivity indicates that the rate of true positives captured by the logistic regression model is (1435/ (1435+11)) 9 0.9924. Likewise, the rate of true negative captured by the logistic regression is (200/ (200+26)) 0.8850, which defines the specificity. Similarly, the logistic regression captured the pos pred values as (1435/ (1435+26)) 0.9822, while it captured the neg pred values as (200/ (200+11)) 0.9479. Lastly, the detection rate tells about the total predicted population, i.e., how much is detected. So, the detection rate in the logistic regression model is (1435/ 1672) 0.8583.

The results of the logistic model, presented and discussed in table 4 and illustration detail of result is present in Figure 8, highlight its effectiveness. The logistic regression achieved an impressive accuracy of 97.7 percent, supported by Recall, Sensitivity, precision, specificity, and F1 score. Furthermore, the LR precision of 0.9822 percent, Recall of 0.9896 percent, F1 score of 0.9859 percent, sensitivity of 0.9897 percent, and specificity of 0.8850 percent further secure the model's high performance.

Table 4 : Evaluation Score of Logistic Regression

Evaluation Measure	Score
Accuracy	0.9779
Precision	0.9822
Recall	0.9896
F1 Score	0.9859
Sensitivity	0.9897
Specificity	0.885



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1435   26
##          1   11  200
##
##                Accuracy : 0.9779
##                  95% CI : (0.9696, 0.984
##     No Information Rate : 0.8648
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9026
##
##  Mcnemar's Test P-Value : 0.02136
##
##             Sensitivity : 0.9924
##             Specificity : 0.8850
##          Pos Pred Value : 0.9822
##          Neg Pred Value : 0.9479
##              Prevalence : 0.8648
##          Detection Rate : 0.8583
##    Detection Prevalence : 0.8738
##       Balanced Accuracy : 0.9387
##
##        'Positive' Class : 0
```

Figure 7 : Confusion Matrix Evaluation of Logistic Regression

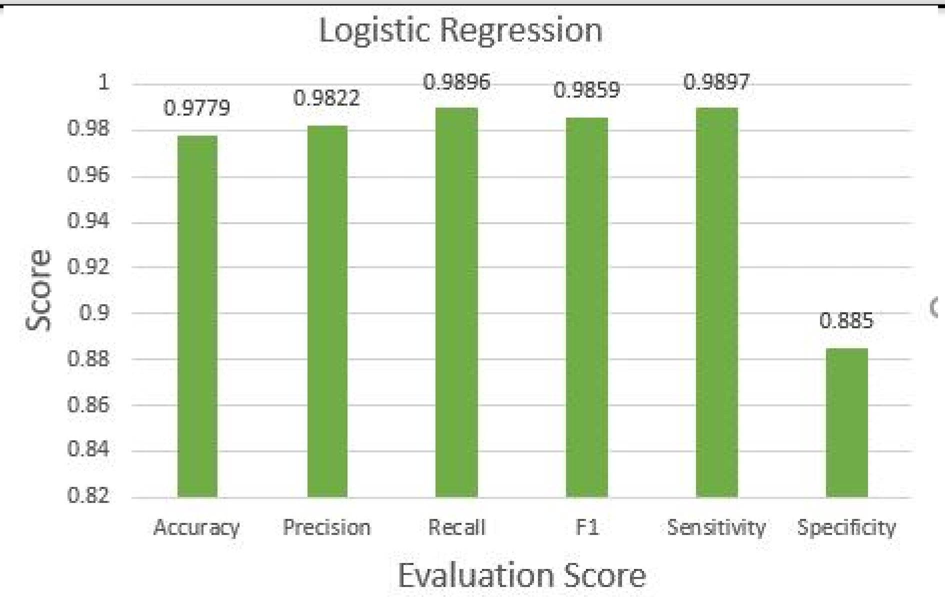


Figure 8 : Evaluation Measure of Logistic Regression

In the results, we produced a confusion matrix with different parameters to interpret the model's significance. The figure in the confusion matrix represented the actual and predicted values. The confusion matrix detail is presented in Figure 9 . In this Figure, at the start, we have a confusion matrix containing 1431 True Negative values out of (1431+15) 1446 actual negative values, and 200 True positive values out of (200+26) 226 actual positive values, which shows how efficiently the NB model is performing. The previous statement can be justified by the accuracy score, which is 97.55, and the model's error rate is 100 - Accuracy, which is 2.45 percent. The accuracy represents the set of correctly classified cases. We can see that there were 1672 cases; out of them, 1431 were correctly classified as 0, and 200 were correctly classified as 1. Thus, out of 1672 classifications, we have 1631 correct classifications, which are 97.5 percent.

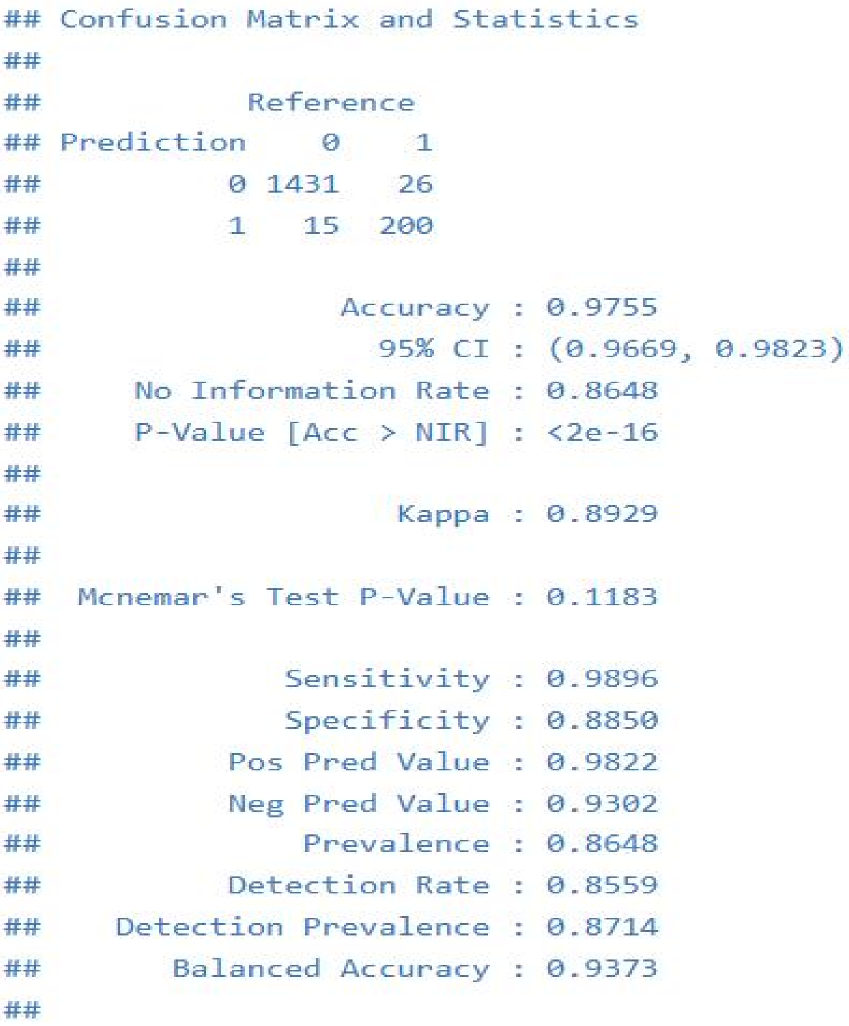
Furthermore, we can see the kappa measure in the confusion matrix, which tests inter-rater reliability and has been frequently used in the literature. In this confusion matrix, we can see that a perfect agreement is found, as kappa produced 0.8929. According to a standard threshold suggested by the researchers, NB is also acceptable according to the kappa.

In addition, we also have other parameters given in the confusion matrix. For example, if we look at the p-value, it shows that the model's accuracy is better. In addition, the sensitivity of the model indicates that the rate of true positive captured by the naïve byes regression model is (1431/ (1431+15)) 0.9896. Likewise, the rate of true negative captured by the naïve byes is (200/ (200+26)) 0.8850, which defines the specificity. Similarly, the naïve bayes captured the pos pred values as (1431/ (1431+26)) 0.9822, while they captured the neg pred values as (200/ (200+15)) 0.9302. Lastly, the detection rate tells about the total predicted population, i.e., how much is detected. So, the detection rate in the logistic regression model is (1431/ 1672) 0.8559. There is a slight difference between the logistic regression and naïve byes regarding detection rate and accuracy. However, logistic regression performs better than naïve byes.

The results of the Naïve byes model are also presented in table 5 as well the illustration detail of model is presented Figure 10. Overall, in the results, with the help of f1 score, sensitivity, precision, specificity, and Recall, the machine learning model Naïve Bayes reached an accuracy score of 97.5 percent, as shown in the table below. NB and LR obtained the same accuracy. In addition, evaluation measurements for NB show a precision of 0.9822 percent, a recall of 0.9896 percent, an F1 score of 0.9859 percent, and a sensitivity of 0.9896 percent, obtaining a specificity of 0.8850 percent score was successful.

Table 5 : Evaluation Results of Naive Bayes

Evaluation Measure	NB Model Score
Accuracy	0.9755
Precision	0.9822
Recall	0.9896
F1 Score	0.9859
Sensitivity	0.9896
Specificity	0.885



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1431   26
##          1   15  200
##
##                Accuracy : 0.9755
##                  95% CI : (0.9669, 0.9823)
##     No Information Rate : 0.8648
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8929
##
##  Mcnemar's Test P-Value : 0.1183
##
##             Sensitivity : 0.9896
##             Specificity : 0.8850
##          Pos Pred Value : 0.9822
##          Neg Pred Value : 0.9302
##              Prevalence : 0.8648
##          Detection Rate : 0.8559
##    Detection Prevalence : 0.8714
##       Balanced Accuracy : 0.9373
##
```

Figure 9 : Confusion Matrix Result of Naive Bayes Model

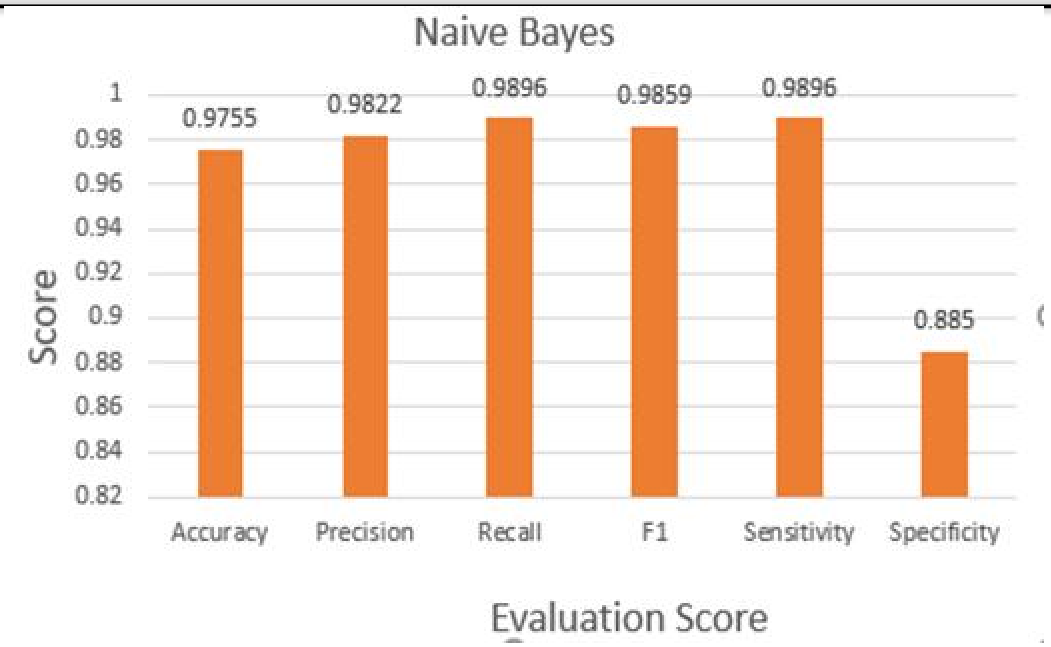


Figure 10 : Evaluation Measure of Naive Bayes Outcomes

The results produced by the random forest model contain a confusion matrix with a set of different parameters that are used to interpret the model's significance. The figure in the confusion matrix represented the actual and predicted values. The confusion matrix is presented in Figure 11. In this Figure, at the start, we have a table of confusion matrix containing 1434 True Negative values out of (1434+12) 1446 actual negative values, and 208 True positive values out of (200+18) 218 actual positive values, which shows how efficiently RF model is performing. The previous statement can be justified by the accuracy score, which is 98.21, and the model's error rate is 100 - Accuracy, which is 1.79 percent. The accuracy represents the set of correctly classified cases. We can see that there were 1672 cases; out of them, 1434 were correctly classified as 0, and 208 were correctly classified as 1. Thus, out of 1672 classifications, we have 1631 correct classifications, which are 98.2 percent.

The confusion matrix also includes the kappa measure, a key indicator of interrater reliability commonly used in the literature. In this instance, the kappa value of 0.9224 indicates a perfect agreement, further validating the random forest model's performance.

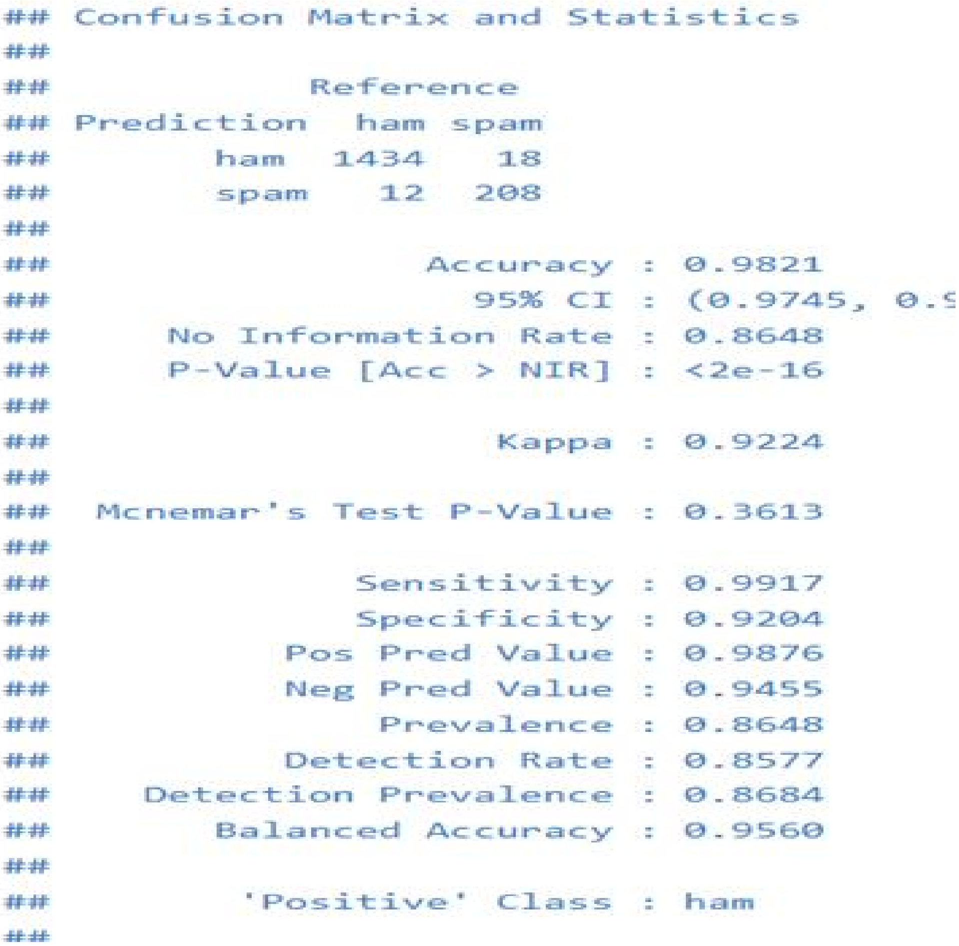
Furthermore, the confusion matrix presents other parameters, including the p-value. This value is a significant indicator of the model's accuracy, providing further insights into its performance.

In addition, the model's sensitivity shows that the rate of true positives captured by the random forest model is (1434/ (1434+12)) 0.9917. Likewise, the rate of true negative captured by the random forest is (208/ (208+18)) 0.9204, which defines the specificity. Similarly, the random forest captured the pos pred values as (1434/ (1434+18)) 0.9876, while the neg pred values were captured as (208/ (208+12)) 0.9455. Lastly, the detection rate tells about the total predicted population, i.e., how much is detected. So, the detection rate in the logistic regression model is (1434/ 1672) 0.8577. In terms of accuracy, RF performs better than NB and LR; however, in terms of detection rate, LR is better than RF, and RF is better than NB, results of RF can be seen in table 6.

The Illustration results of the RF model are presented in Figure 12 and given detail of outcomes in Table no 06. According to the data presented in the following table, the machine learning model known as Random Forest achieved an accuracy of 98.1 percent when evaluated based on its f1 score, sensitivity, precision, specificity, and recall. In addition, the evaluation measurements for SVM reveal a precision of 0.9869 percent, a recall of 0.9917 per cent, an F1 score of 0.9873 percent, and a sensitivity of 0.9917 percent, and they were successful in getting a specificity score of 0.9159 percent.

Table 6 : Evaluation Results of Random Forest

Evaluation Measure	RF Model Score
Accuracy	0.9815
Precision	0.9869
Recall	0.9917
F1 Score	0.9893
Sensitivity	0.9917
Specificity	0.9159



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  ham spam
##       ham  1434   18
##       spam   12  208
##
##                Accuracy : 0.9821
##                  95% CI : (0.9745, 0.9
##     No Information Rate : 0.8648
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9224
##
##  Mcnemar's Test P-Value : 0.3613
##
##             Sensitivity : 0.9917
##             Specificity : 0.9204
##          Pos Pred Value : 0.9876
##          Neg Pred Value : 0.9455
##              Prevalence : 0.8648
##          Detection Rate : 0.8577
##    Detection Prevalence : 0.8684
##       Balanced Accuracy : 0.9560
##
##        'Positive' Class : ham
##
```

Figure 11 : Evaluation of Confusion Matrix RF

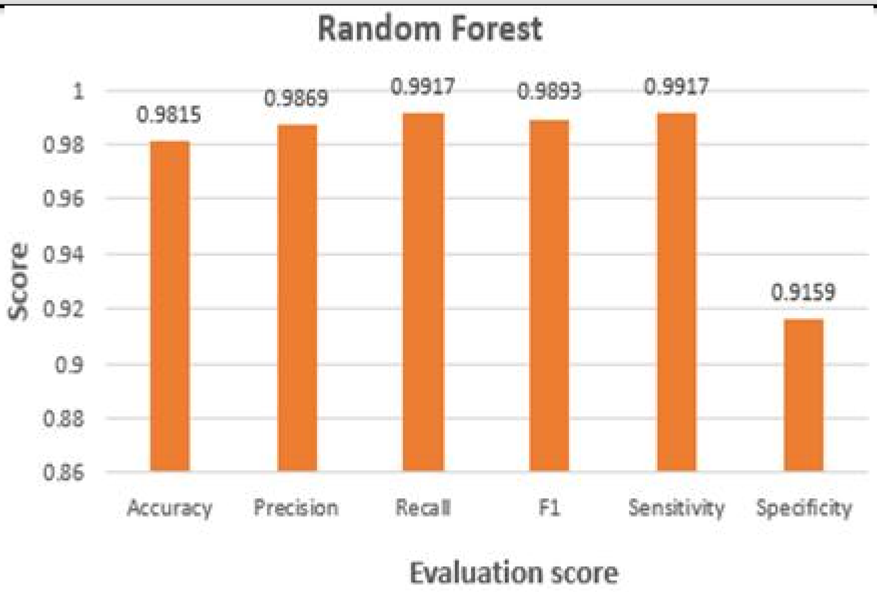


Figure 12 : RF Outcomes Evaluation

We found a confusion matrix with functional parameters for validating the model in the experimental results. This table in the confusion matrix represents the actual and predicted values, and this matrix helps interpret different aspects of the classifier's quality. The confusion matrix is presented in Figure 13. In this Figure, the confusion matrix is given at the start, containing 1435 True Negative values out of (1435+11) 1446 actual negative values, and 200 True positive values out of (200+26) 226 actual positive values, which shows how efficiently SVM model is working. The accuracy score of 97.79 can justify the previous statement and the model's error rate, which is 2.21 percent. The accuracy represents the set of correctly classified cases. We can see that there were 1672 cases; out of them, 1435 were correctly classified as 0, and 200 were correctly classified as 1. Thus, out of 1672 classifications, we have 1635 correct classifications, which are 97.7%.

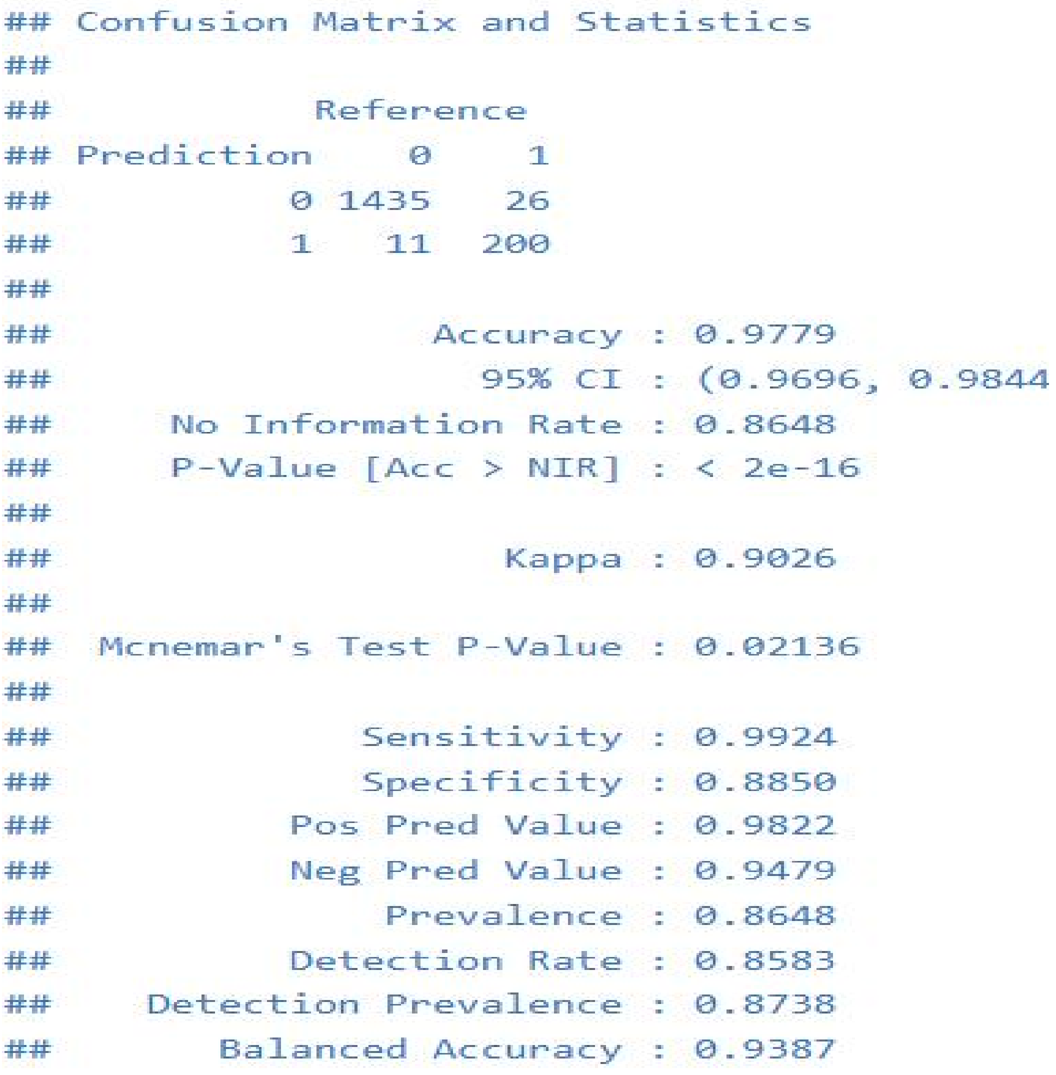
Furthermore, the kappa measure in the confusion matrix is another statistical method to test inter-rater reliability and has been frequently used in the literature. The range of the kappa value is -1 to +1, where 0 shows the amount of agreement expected from a random chance, and 1 shows a perfect agreement. The author of the Kappa measure suggested that the value less than or equal to 0 indicates no agreement, the range 0.01 – 0.20 shows a slight agreement, the range 0.21 – 0.40 shows a fair agreement, the range 0.41 - 0.60 shows a moderator and 0.61 – 0.81 shows a substantial and 0.81-1.00 shows a perfect agreement. Here, in this confusion matrix, we can see that an ideal deal is found. In this research, many researchers suggest that 80 percent is acceptable. Therefore, our model is permissible according to the kappa.

Apart from this, other statistical parameters are also given in the confusion matrix. Furthermore, if we look at the p-value, it indicates that the accuracy of the classifier is more acceptable. In addition, the sensitivity of the model indicates that the rate of true positives captured by the logistic regression model is (1435/ (1435+11)) 90.9924. Likewise, the rate of true negative captured by the logistic regression is (200/ (200+26)) 0.8850, which defines the specificity. Similarly, the logistic regression captured the pos pred values as (1435/ (1435+26)) 0.9822, while it captured the neg pred values as (200/ (200+11)) 0.9479.

The illustration results of the support vector machine model are presented in Figure 14, and detail result of SVM is given in Table 7. Outcomes of Support Vector Machine are brief in the table that can be found further down on this side. With the help of sensitivity, specificity, precision, Recall, and f1 score, the machine learning model SVM could reach an accuracy of 97.7 percent, as shown in the table below. SVM and LR obtain the same accuracy. In addition, evaluation measurements for SVM show a precision of 0.9822 percent, a recall of 0.9924 percent, an F1 score of 0.9873 percent, and a sensitivity of 0.9924 percent, obtaining a specificity of 0.8850 percent score was successful.

Table 7 : Evaluation Results of SVM

Evaluation Measure	SVM Model Score
Accuracy	0.9779
Precision	0.9822
Recall	0.9924
F1 Score	0.9873
Sensitivity	0.9924
Specificity	0.885



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1435   26
##          1   11  200
##
##                Accuracy : 0.9779
##                  95% CI : (0.9696, 0.9844
##     No Information Rate : 0.8648
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9026
##
##  Mcnemar's Test P-Value : 0.02136
##
##             Sensitivity : 0.9924
##             Specificity : 0.8850
##          Pos Pred Value : 0.9822
##          Neg Pred Value : 0.9479
##              Prevalence : 0.8648
##          Detection Rate : 0.8583
##    Detection Prevalence : 0.8738
##       Balanced Accuracy : 0.9387
```

Figure 13 : Confusion Matrix Analysis SVM

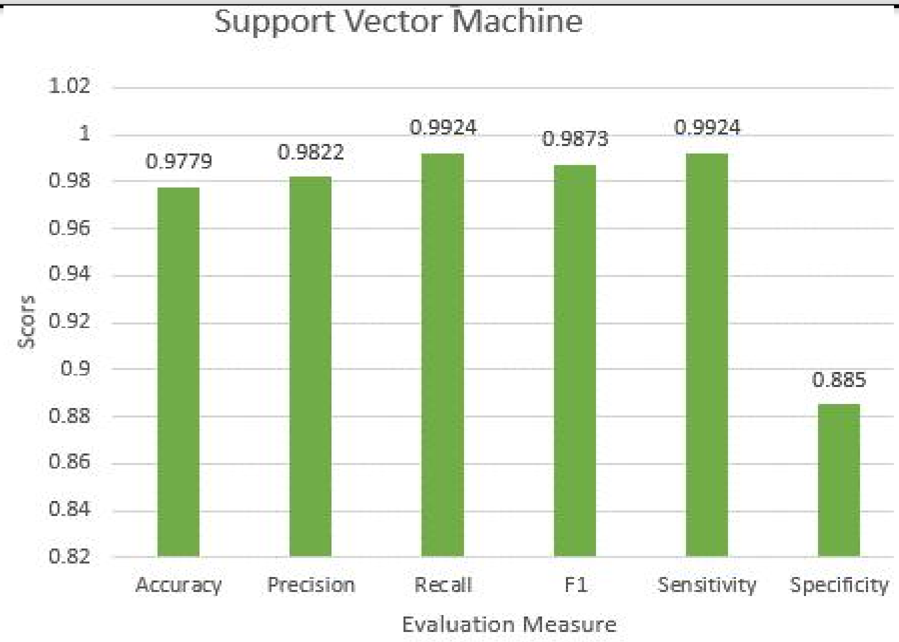


Figure 14 : Outcomes Evaluation Analysis of SVM

In the results, we produced a confusion matrix with functional parameters to validate the model. This figure in the confusion matrix represents the actual and predicted values, as well as this matrix helps interpret different aspects of the quality of the classifier. The confusion matrix is presented in Figure 15. In this Figure, the graph diagram of confusion matrix is given at the start, containing 1435 True Negative values out of (1436+10), 1446 actual negative values, and 206 True positive values out of (206+20) 226 actual positive values, which shows how efficiently GBM model is working. The accuracy score of 98.21 percent can justify the previous statement and the model's error rate, which is 1.79 percent. The accuracy represents the set of correctly classified cases. We can see that there were 1672 cases; out of them, 1436 were correctly classified as 0, and 206 were correctly classified as 1. Thus, out of 1672 classifications, we have 1642 correct classifications, which is 97.7 percent. This high accuracy of the GBM model instils confidence in its performance.

Furthermore, the kappa measure in the confusion matrix is another statistical method to test inter-rater reliability and has been frequently used in the literature. The range of the kappa value is -1 to +1, where 0 shows the amount of agreement expected from a random chance, and 1 shows a perfect agreement. The author of the Kappa measure suggested that the value less than or equal to 0 indicates no agreement, the range 0.01 – 0.20 shows a slight agreement, the range 0.21 – 0.40 shows a fair agreement, the range 0.41 - 0.60 shows a moderator and 0.61 – 0.81 shows a substantial and 0.81-1.00 shows a perfect agreement. Here, in this confusion matrix, we can see that the GBM model's kappa value of 0.9218 indicates a fair agreement, making its performance reliable. In this research, many researchers suggest that 80 percent is acceptable. Therefore, our model is permissible according to the kappa by producing 0.9218.

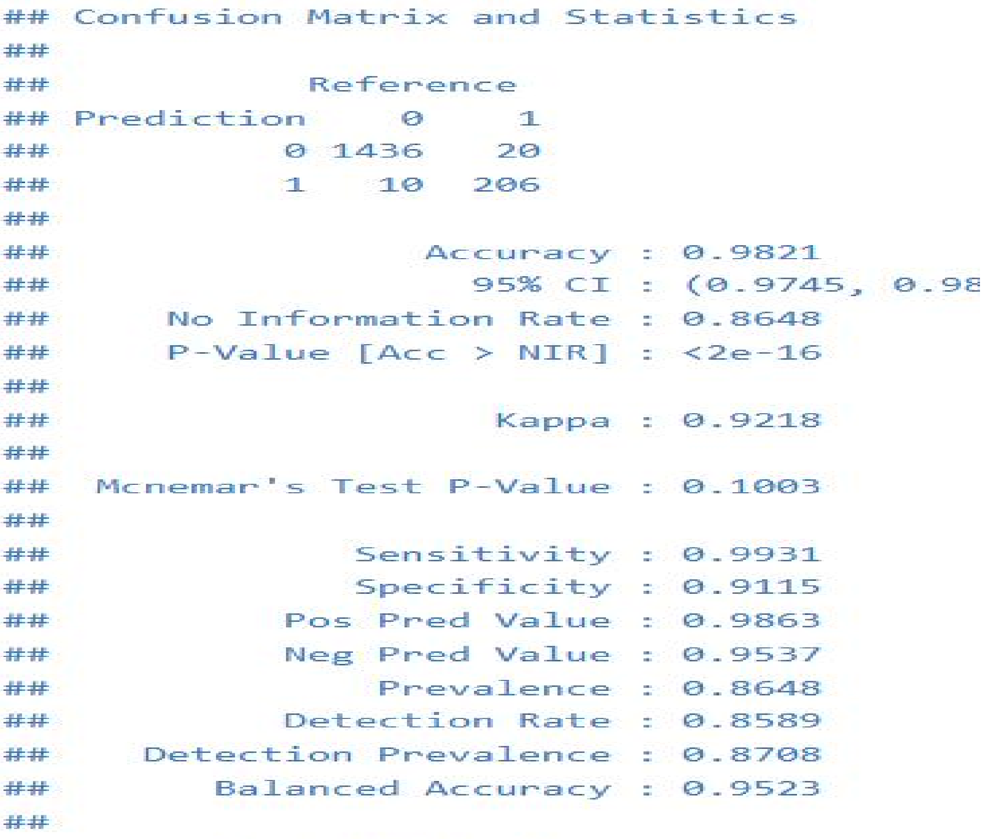
Apart from this, other statistical parameters are also given in the confusion matrix. These parameters, such as the p-value, sensitivity, specificity, pos pred values, neg pred values, and detection rate, provide a comprehensive understanding of the model's performance. Furthermore, if we look at the p-value, it shows that the model's accuracy is better. In addition, the sensitivity of the model indicates that the rate of true positives captured by the logistic regression model is (1436/ (1436+10)) 0.9931. Likewise, the rate of true negative captured by the logistic regression is (206/ (206+20)) 0.9115, which defines the specificity. Similarly, the logistic regression captured the pos pred values as (1436/ (1436+20)) 0.9863, while it captured the neg pred values as (206/ (206+110)) 0.9537. Lastly, the detection rate tells about the total predicted population, i.e., how much is detected. So, the detection rate in the logistic regression model is (1436/ 1672) 0.8589. Compared to the previous models, GBM performed well in terms of accuracy and detection rate.

The results of the Gradient Boosting Machine are summarized in Table 8 and detail illustration result of GBM model is given in figure 16, located further down on this side. According to the data presented in the following table, the machine learning model known as GBM achieved an accuracy of 98.2 percent when evaluated based on its sensitivity, specificity, precision, recall, and f1 score.

In addition, the evaluation measurements for SVM reveal a precision of 0.9863 percent, a recall of 0.9931 percent, an F1 score of 0.9897 percent, and a sensitivity of 0.9931 percent. They were also successful in getting a specificity score of 0.9115 percent. Overall, GBM achieved the highest accuracy.

Table 8 : Evaluation Results of GBM

Evaluation Measure	GBM Model Score
Accuracy	0.9821
Precision	0.9863
Recall	0.9931
F1 Score	0.9897
Sensitivity	0.9931
Specificity	0.9115



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1436   20
##          1   10  206
##
##                Accuracy : 0.9821
##                  95% CI : (0.9745, 0.98
##     No Information Rate : 0.8648
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9218
##
##  Mcnemar's Test P-Value : 0.1003
##
##             Sensitivity : 0.9931
##             Specificity : 0.9115
##          Pos Pred Value : 0.9863
##          Neg Pred Value : 0.9537
##              Prevalence : 0.8648
##          Detection Rate : 0.8589
##    Detection Prevalence : 0.8708
##       Balanced Accuracy : 0.9523
##
```

Figure 15 : Confusion Matrix Analysis of GBM

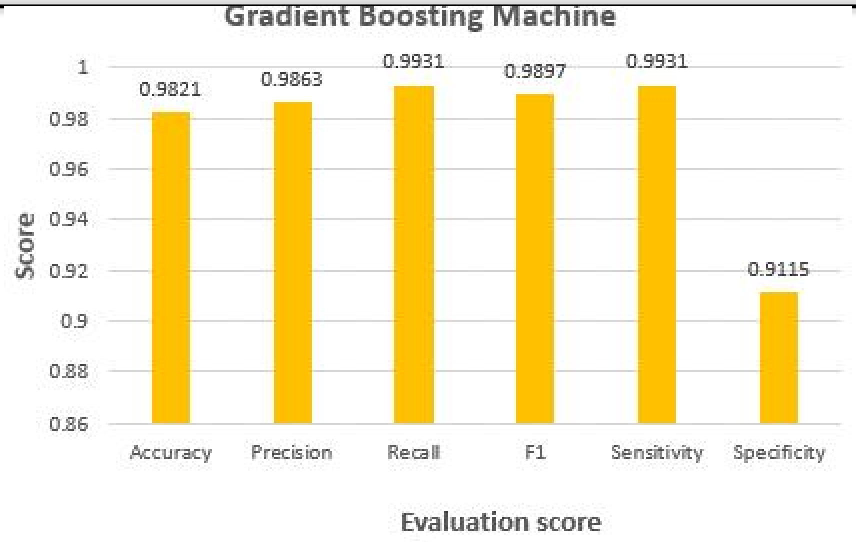


Figure 16 : Result Evaluation Analysis of GBM

Light GBM has a wide range of tunable parameters, which we will attempt to optimize using Optuna, a hyper-parameter optimization framework. Using this model, we have predicted the SMS class (i.e., either spam or ham) using SMS metadata (i.e., extracted binary and numeric features). Where 'is-spam-binary' is used as a dependent variable, the remaining features are used as independent variables.

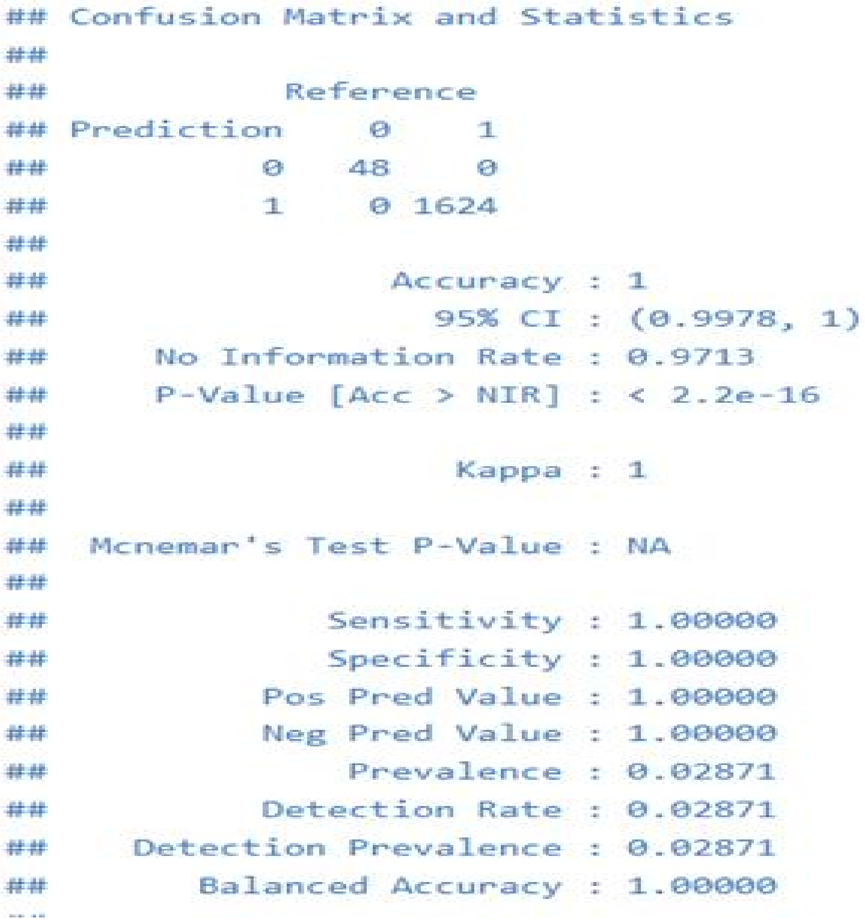
The results produced by the random forest model contain a confusion matrix with a set of different parameters. The confusion matrix detail is presented in Figure 17. In this Figure, at the start, we have a table of confusion matrix containing 1624 True Negative values out of (1624+0) 1624 actual negative values, and 48 True positive values out of (48+0) 48 actual positive values, which shows how efficiently LGBM model is performing. The accuracy score of the model is 100 percent, and the model's error rate is 0 percent. The accuracy represents the set of correctly classified cases. We can see that there were 1672 cases; out of them, 1624 were correctly classified as 1, and 48 were correctly classified as 0. Thus, out of 1672 classifications, we have 1672 correct classifications that are 100 percent.

In addition, we can see that kappa produced 1, which shows a perfect agreement. We also have sensitivity and specificity as 1, which shows the model's perfection. Apart from this, we can see that the model produced pos pred value and neg pred value as 1. However, the detection rate of this model is 0.0287, which is too low compared to all the machine learning models in this research.

The illustration results of the Light GBM model are presented in Figure 18, as the detail result of model given in Table 8. The table that can be found further down on this side summarises the results obtained from the Light Gradient Boosting Machine. When the machine learning model known as LGBM was evaluated based on its sensitivity, specificity, precision, recall, and f1 score, the data presented shows that it reached an accuracy of 100 percent. In all Machine learning models used in current research, when compared to LGBM, the LGBM achieved the highest accuracy. This information is presented in the table that comes after the table that presents the data. In addition, the evaluation measurements for LGBM validate an accuracy of 100 percent, a recall of 100 percent, an F1 score of 100 percent, and a sensitivity of 100 percent. Furthermore, they were successful in obtaining a specificity score of 100 percent, see table 9.

Table 9 : Evaluation Outcomes Measures for LGB Machine

Evaluation Measure LGBM	Accuracy %
Accuracy	100
Precision	100
Recall	100
F1 Score	100
Sensitivity	100
Specificity	100



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0   48    0
##          1    0 1624
##
##                Accuracy : 1
##                  95% CI : (0.9978, 1)
##     No Information Rate : 0.9713
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.00000
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 1.00000
##              Prevalence : 0.02871
##          Detection Rate : 0.02871
##    Detection Prevalence : 0.02871
##       Balanced Accuracy : 1.00000
```

Figure 17 : Confusion Matrix of LGBM

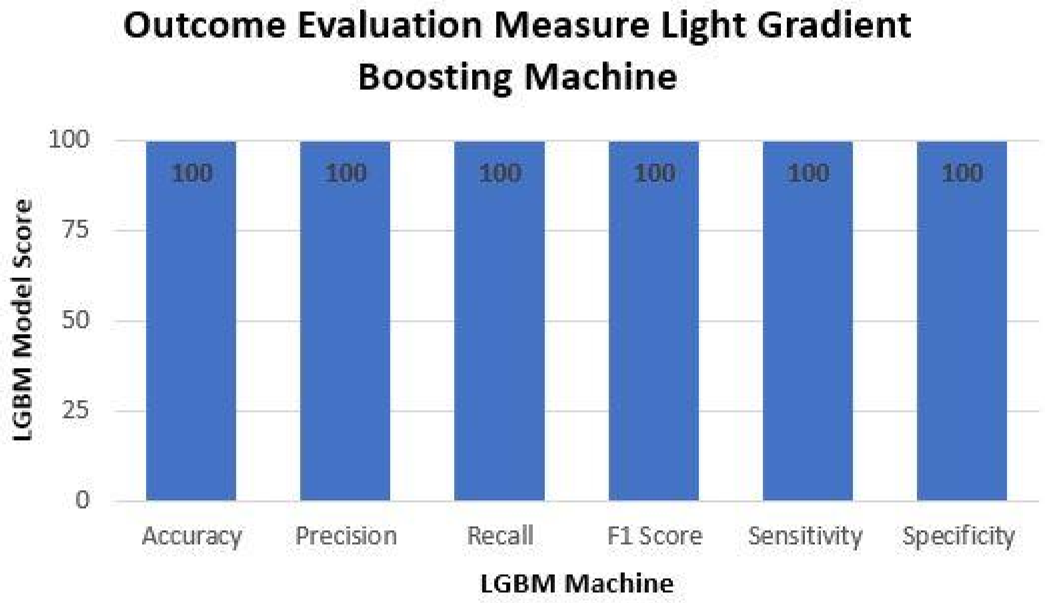


Figure 18 : Evaluation Measure Light Gradient Boosting Machine

4.4 : Research outcomes Discussion of All State of Art the Classifiers

In Our Research Study, we used six state-of-the-art trained machine learning algorithms. The six algorithms are Logistic Regression(LR), Random Forest(RF), Support vector machine(SVM), and Gradient Boosting Machine (GBM), Light gradient boosting machine(LGBM). These models are used to classify SMS spam detection. Both logistic regression and SVM models give an accuracy score of 97.79 percent accuracy. In contrast, Naive Bayes gives an accuracy score for the classification of SMS spam of 97.55 accuracy score. The Random forest gives an accuracy score of 98.15, GBM gives an accuracy score of 98.21, and the Light gradient boosting machine (LGBM) has the highest accuracy score of 100 in the classification of SMS spam detection with an error rate of 0.

The study evaluation results show that all our applied machine-learning models perform well, and Graph 19 explains the details of all the models.

The Naive Bayes classifier gives the lowest accuracy score of 97.55, while LR and SVM give a reasonable score of 97.55. The GBM classifier produced the second-highest accuracy score, 98.21, and RF gave an accuracy score 98.15. Our projected Model produced the Maximum accuracy score of 100 with the lowest error rate of 0, which defeats all the other comparable models.

The latter performed well and achieved the highest accuracy by 100 percent. Table 10 compares all the Models.

Table 10 : Comparison Analysis of Applied Machine Learning Models

Machine Learning Models	Accuracy %
LR	97.79
RF	98.15
SVM	97.79
NB	97.55
GBM	98.21
LGBM	100.0

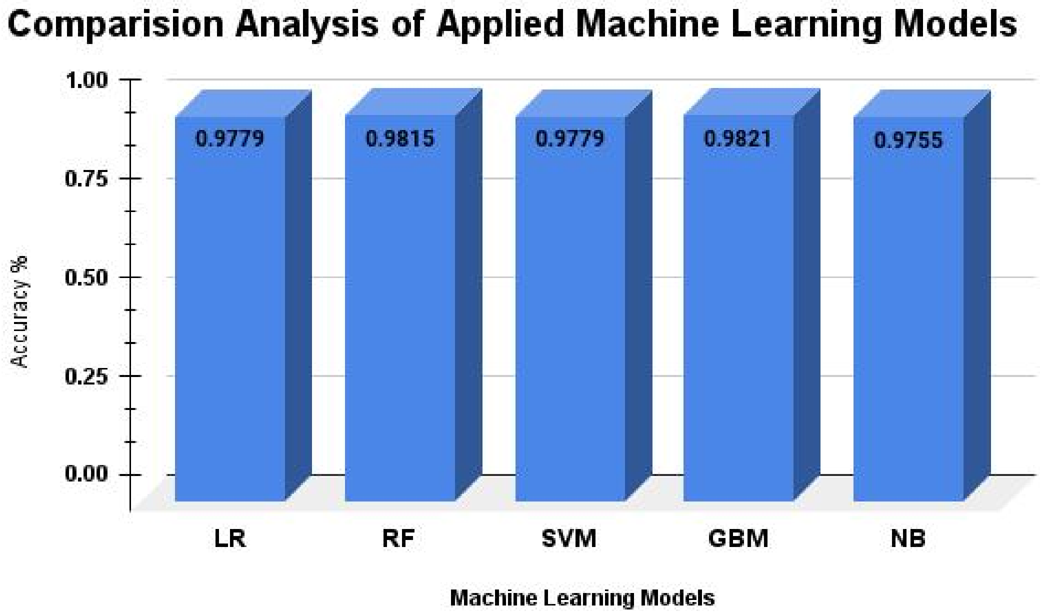


Figure 19 : Comparison Analysis of Applied Machine Learning Models

4.5 : Result Comparison Analysis to the state of the art study

We conducted the comparison analysis by employing our dataset with previous research. The earlier study used the Multinomial Naïve Bayes (MNB) approach and achieved a maximum accuracy of 98.50, an F-1 Score of 98, and an AUC of 97. The evaluation parameters are the year, approach, forecasted approach, F-1 score, and accuracy score. Our analysis showed that the proposed model of our study, a light gradient boosting machine (LGBM), outperformed the previous research. Our technique, the LGBM technique, delivered the most accurate outcomes.

Ref	Year	Approach	Forecasted Approach	F-1 Score	Precision	Recall	Accuracy
[15]	2023	Machine learning	Multinomial Naïve Bayes	97.7	-	-	98.50
Proposed	2025	Machine learning	Light Gradient Boosting Machine (LGBM)	100	100	100	100

5 - Conclusion

This study is about detecting spam and ham in SMS messages. Experiments were conducted on various classifiers to evaluate their potential as benchmark machine-learning models for detecting SMS spam. The accuracy of these classical classifiers in categorizing spam in the dataset was excellent. The features are extracted using feature extraction, which uses eleven distinct characteristics, including spam count, ham count, and spam binary. Other features include char count, number, URL, date, dollar, emoticon, email, and phone. To accomplish this, a dataset that consists of the text of the SMS Spam is obtained. We use this data and transform it into two categories, ham and spam, to use it for the target class. After that, the dataset is split into two sets: the training set, which comprises 70 percent of the data, and the test set, which shall consist of 30 percent. We use the training set that we used to train the learning models logistic regression, Random Forest, Support vector machine, naive Bayes, Gradient boosting machine, light Gradient boosting machine, and the subsequent training of learning models to evaluate the performance of these learning models by passing test data to the trained models that they were trained on. The primary focus of this research is the accuracy of models and whether or not SMS messages should be classified as spam or ham. The SMS spam dataset is described on Kaggle. The table that can be found above displays the outcomes of different classification models that were applied to the SMS spam data set that contained features. After doing the research and making the necessary comparisons, we concluded that LGBM is the best classifier and attained the most remarkable accuracy score of 100 percent.

Even though the outcomes of the experiments carried out for this study have shown that the proposed model for the detection of SMS spam is an improvement over some of the earlier approaches to the problem of SMS spam detection, we continue to have the impression that the model we proposed holds a significant amount of potential that has not yet been utilized. To begin, given that the datasets that are currently available only contain thousands of messages, we plan to extend our SMS spam Detection model in the future to a larger dataset using deep learning approaches such as convolutional neural network (CNN), long-term short-term memory (LSTM), Ensemble Neural Network(ENN), and Recurrent neural network (RNN), as well as additional messages or even other types of content, to improve its overall performance and then classifying it accordingly. This will be accomplished in the future.

REFERENCES

[1] Rahman, S. E., & Ullah, S. (2020a). Email spam detection using bidirectional long short term memory with convolutional neural network. 2020 IEEE Region 10 Symposium (TENSYMP), 1307–1311.

[2]Jain, H., & Maurya, R. K. (2022). A review of sms spam detection using features selection. 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), 101–106.

[3]Suleiman, D., Al-Naymat, G., & Itriq, M. (2020). Deep sms spam detection using h2o platform. International Journal, 9(5).

[4] Ramanujam, E., Shankar, K., & Sharma, A. (2022). Multi-lingual spam sms detection using a hybrid deep learning technique. 2022 IEEE Silchar Subsection Conference (SILCON), 1–6.

[5] Jain, G., Sharma, M., & Agarwal, B. (2019a). Optimizing semantic lstm for spam detection. International Journal of Information Technology, 11, 239–250.

[6] Roy, S. S., Sinha, A., Roy, R., Barna, C., & Samui, P. (2018). Spam email detection using deep support vector machine, support vector machine and artificial neural network. Soft Computing Applications: Proceedings of the 7th International Workshop Soft Computing Applications (SOFA 2016), Volume 2 7, 162–174.

[7] Svadasu, G., & Adimoolam, M. (2022). Spam detection in social media using artificial neural network algorithm and comparing accuracy with support vector machine algorithm. 2022 International conference on business analytics for technology and security (ICBATS), 1–5.

[8] Jain, G., Sharma, M., & Agarwal, B. (2019b). Spam detection in social media using convolutional and long short term memory neural network. Annals of Mathematics and Artificial Intelligence, 85(1), 21–44.

[9] Roy, P. K., Singh, J. P., & Banerjee, S. (2020). Deep learning to filter sms spam. Future Generation Computer Systems, 102, 524–533.

[10] Jain, G., Sharma, M., & Agarwal, B. (2019c). Optimizing semantic lstm for spam detection. International Journal of Information Technology, 11, 239–250.

[11] Liu, X., Lu, H., & Nayak, A. (2021). A spam transformer model for sms spam detection. IEEE Access, 9, 80253–80263.

[12] Rahman, S. E., & Ullah, S. (2020b). Email spam detection using bidirectional long short term memory with convolutional neural network. 2020 IEEE Region 10 Symposium (TENSYMP), 1307– 1311.

[13] Gadde, S., Lakshmanarao, A., & Satyanarayana, S. (2021). Sms spam detection using machine learning and deep learning techniques. 2021 7th international conference on advanced computing and communication systems (ICACCS), 1, 358–362.

[14] Hossain, S. M. M., Kamal, K. M. A., Sen, A., & Sarker, I. H. (2023). Tf-idf feature-based spam filtering of mobile sms using a

machine learning approach. In Applied intelligence for industry 4.0 (pp. 162–175). Chapman; Hall/CRC.

[15] Sheikhi, S., Kheirabadi, M. T., & Bazzazi, A. (2020). An effective model for sms spam detection using content-based features and averaged neural network. International Journal of Engineering, 33(2), 221–228.

[16] Sharmin, S., & Zaman, Z. (2017). Spam detection in social media employing machine learning tool for text mining. 2017 13th international conference on signal-image technology & internetbased systems (SITIS), 137–142.

[17] Jain, G., Sharma, M., & Agarwal, B. (2017). Spam detection on social media text. International Journal of Computer Science and Engineering, 5.

[18] Choudhary, N., & Jain, A. K. (2017). Towards filtering of sms spam messages using machine learning based technique. Advanced Informatics for Computing Research: First International Conference, ICAICR 2017, Jalandhar, India, March 17–18, 2017, Revised Selected Papers, 18–30.

[19] Deshmukh, R., et al. (2021). Performance comparison for spam detection in social media using deep learning algorithms. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(1S), 193–201.

[20] Jain, T., Garg, P., Chalil, N., Sinha, A., Verma, V. K., & Gupta, R. (2022). Sms spam classification using machine learning techniques. 2022 12th international conference on cloud computing, data science & engineering (confluence), 273–279.

[21] Sharma, N. (2022). A methodological study of sms spam classification using machine learning algorithms. 2022 2nd International Conference on Intelligent Technologies (CONIT), 1–5.

[22] Https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset. (n.d.).

[23] Robles-Velasco, A., Cortes, P., Mu ´ nuzuri, J., & Onieva, L. (2020). Prediction of pipe failures in water supply networks using logistic regression and support vector classification. Reliability Engineering & System Safety, 196, 106754.

[24] Robles-Velasco, A., Cortes, P., Mu ´ nuzuri, J., & De Baets, B. (2023). Prediction of pipe failures in water supply networks for longer time periods through multi-label classification. Expert Systems with Applications, 213, 119050.

[25] Gata, W., Basri, H., Hidayat, R., Patras, Y. E., Baharuddin, B., Fatmasari, R., Tohari, S., & Wardhani, N. K. (2019). Algorithm implementations naıve bayes, random forest. c4. 5 on online gaming for learning achievement predictions. 2nd International Conference on Research of Educational Administration and Management (ICREAM 2018), 1–9.

[26] Qadri, A. M., Hashmi, M. S. A., Raza, A., Zaidi, S. A. J., & ur Rehman, A. (2024). Heart failure survival prediction using novel transfer learning based probabilistic features. PeerJ Computer Science, 10, e1894.

[27] Haider, M., Hashmi, M. S. A., Raza, A., Ibrahim, M., Fitriyani, N. L., Syafrudin, M., & Lee, S. W. (2024). Novel ensemble learning algorithm for early detection of lower back pain using spinal anomalies. Mathematics, 12(13), 1955. https://doi.org/10.3390/math12131955

[28] Qadri, A. M., Raza, A., Munir, K., & Almutairi, M. S. (2023). Effective feature engineering technique for heart disease prediction with machine learning. IEEE Access, 11, 56214–56224.

[29] Farooq, H., Hashmi, M. S. A., Khan, T. F., Hafeez, Q., & Mohsin, M. (2024). Intelligent emergency vehicle sound classification for public safety. Kashf Journal of Multidisciplinary Research, 1(12), 141–152. https://doi.org/10.71146/kjmr161

[30] Wang, H. (2022a). Forecasting credit card defaults using light gradient boosting machine with dart algorithm. Proceedings

of the 2022 5th Artificial Intelligence and Cloud Computing Conference, 207–212.

[31] Wang, H. (2022b). Forecasting credit card defaults using light gradient boosting machine with dart algorithm. Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference, 207–212.